

State Space Model of a Light Weight Flexible Telescope

A Thesis

Presented to the Faculty of

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In Mechanical Engineering

By

Chi-Yeh Hsu

August 2010

©2010
Chi-Yeh Hsu
All Rights Reserved

Committee Membership

Title: State Space Model of Light Weight Flexible Telescope

Author: Chi-Yeh Hsu

Date Submitted: August, 2010

Committee Chair: Dr. John Ridgely
Assistant Professor
Cal Poly Mechanical Engineering

Committee Member: Dr. Charles Birdsong
Associate Professor
Cal Poly Mechanical Engineering

Committee Member: Dr. Xi Wu
Assistant Professor
Cal Poly Mechanical Engineering

Abstract

State Space Model of a Light Weight Flexible Telescope

Chi-Yeh Hsu

The accuracy of optical telescope in the past was dominated by the construction of the telescope body and the precision of optical surface. This thesis fits into a larger project with a goal of controlling a flexible telescope in order to help a cheaper telescope achieve higher accuracy. In order to understand the internal motion of telescope, the goal of this thesis is to build a state space model of a two-degree-of-freedom telescope using a known star location as feedback. The model presented in this thesis may be used for the telescope simulation and the control design in the future. This project considers a light weight flexible structured telescope and uses state space formulation to model dynamic motion of the telescope since state space model is better for modeling and analyzing a multiple-inputs and multiple-outputs system. A one-degree-freedom experimental test stand is built to generate a series of telescope motion data in order to examine whether the telescope simulation model can represent a real telescope system. To record the telescope motion, a wheel encoder mounted on the azimuth motor and a LED light density sensor installed at the end of telescope body are used as feedbacks. The results show that the state space model can simulate many system dynamic behaviors as shown in the experimental result; however, it is not feasible to obtain all of telescope's physical parameters from the system feedbacks.

Table of Contents

List of Figures	vii
List of Tables.....	viii
List of Equations	ix
Chapter 1: Introduction.....	1
1.1: Background	1
1.2: Star Tracking and Telescope Control	4
1.2.1 Altazimuth Tracking	4
1.2.2: Open Loop and Closed Loop Position Control	6
1.2.3: Disturbances	8
1.2.4: Stability Criteria	1 0
1.3: Project Overview and Objective.....	1 1
Chapter 2: Model of Flexible Telescope	1 3
2.1: Telescope System	1 3
2.1.1: Structure Components	1 4
2.1.2: Control Inputs	1 6
2.1.3: Sensor Feedback.....	1 6
2.2: Model of External Disturbances.....	1 7
2.2.1: Gravity	1 7
2.2.2: Wind	1 8
2.3: Azimuth System.....	1 9
2.3.1: Model of Azimuth System.....	1 9
2.3.2: Azimuth Model State Equations.....	2 2
2.3.3: State-space Model of Azimuth System	2 6
2.4: Altitude System	3 1
2.4.1: Model of Altitude System	3 1
2.4.2: Altitude Model State Equations	3 4
2.4.3: State-space Model of Altitude System	3 9
2.5: Matlab Simulation	4 4
2.5.1: Simulink Model.....	4 4
2.5.2: Simulation Output	4 8
Chapter 3: Azimuth Experimental Telescope	5 2

3.1: Experiment Set Up	5	2
3.1.1: Telescope Structure	5	2
3.1.2: Software and Data Acquisition Interface.....	5	4
3.1.3: Guide Star Feedback Simulation	5	5
3.2: Experiments Result.....	5	8
3.2.1: Light Feedback	5	9
3.2.2: PI Control	6	1
3.2.3: Tap Test	6	2
3.2.4: Feedback Control Using Light Feedback	6	5
Chapter 4: Simulation Result Analysis	6	9
4.1: Turned Simulation model.....	6	9
4.1.1: PI Control Tuning	6	9
4.1.2: Frequency Response of Simulation Model	7	0
4.2: Controllability and Observability Analysis	7	2
Chapter 5: Conclusion.....	7	4
5.1: Light Weight Flexible Telescope Model.....	7	4
5.2: Experimental Test Stand	7	4
5.3: Control Analysis.....	7	5
5.4: Current Project State and Future Work.....	7	6
Reference	7	7
Appendix A: Derivation of Azimuth Model	7	8
Appendix B: Disturbance Model.....	8	6
Appendix C: Derivation of Altitude Model	8	9
Appendix D: Matlab Simulation Model	9	9
Appendix E: Matlab Plot Functions	1	7
Appendix F: OTA Calculation.....	1	7
Appendix G: System Parameters	1	7

List of Figures

Figure 1: Stationary Telescope - Hooker telescope at Mount Wilson Observatory ...	2
Figure 2: Homemade telescope	3
Figure 3: Altitude (h) and Azimuth (A) coordination.....	5
Figure 4: Telescope Open Loop Control flow chart.....	7
Figure 5: Telescope Closed Loop Control flow chart.....	8
Figure 6: Generalized Telescope System.....	1 3
Figure 7: Telescope System Components.....	1 5
Figure 8: Azimuth Telescope Model.....	2 2
Figure 9: Azimuth Telescope Model Tree Diagram	2 3
Figure 10: Star Image Displacement	2 8
Figure 11: Altitude Telescope Model	3 4
Figure 12: Altitude Telescope Model Tree Diagram.....	3 6
Figure 13: State-space subsystem.....	4 5
Figure 14: Wind model for altitude system	4 6
Figure 15: Gravity model	4 6
Figure 16: Full telescope in Simulink.....	4 7
Figure 17: PID controller	4 8
Figure 18: Result of azimuth simulation	4 9
Figure 19: Result of altitude Simulation.....	5 0
Figure 20: pointing path and star image feedback.....	5 1
Figure 21: Experimental Setup of experimental telescope.....	5 3
Figure 22: Motomatic Station	5 4
Figure 23: SC-2075 DAQ Interface.....	5 5
Figure 24: Simulated Star Source	5 6
Figure 25: Star Image senor used on Test Stand.....	5 6
Figure 26: Star image feedback simulation sensor situation	5 7
Figure 27: Light response from Simulation	5 8
Figure 28: Telescope test	5 9
Figure 29: Star Image feedback reading.....	6 0
Figure 30: Telescope control response (PID control)	6 2
Figure 31: OTA response to tap test.....	6 4
Figure 32: Frequency response of telescope to tap test	6 5
Figure 33: PI control using star image feedback.....	6 7
Figure 34: Turned Simulation Result	7 0
Figure 35: Frequency response of simulation model.....	7 1

List of Tables

Table 1: Telescope system components	1 5
Table 2: Control Inputs of telescope	1 6
Table 3: Sensor Feedback of telescope.....	1 7
Table 4: Modeling Parameters for Azimuth Model	2 1
Table 5: Modeling parameters for Altitude Model (Part1)	3 3
Table 6: Modeling parameters for Altitude Model (Part2)	3 4
Table 7: Natural frequency of components	7 2
Table 8: Controllability and Observability Analysis	7 2

List of Equations

Equation 1: Gravitational disturbance	1 8
Equation 2: Wind disturbance	1 8
Equation 3: States of azimuth system	2 3
Equation 4: State space equation of azimuth model	2 4
Equation 5: Torque transformation of azimuth model.....	2 4
Equation 6: Transformed azimuth model States	2 5
Equation 7: Transformed state equation for azimuth model.....	2 6
Equation 8: State Space Model for Azimuth	2 7
Equation 9: Azimuth Model Input.....	2 7
Equation 10: Azimuth Model Output	2 7
Equation 11: Star Image Displacement Calculation	2 8
Equation 12: Reduced Star Image Displacement Equation.....	2 8
Equation 13: Azimuth Model disturbance Source.....	2 8
Equation 14: State (A) matrix for azimuth model	2 9
Equation 15: Input (B) matrix for azimuth model	2 9
Equation 16: Output (C) matrix for azimuth model	3 0
Equation 17: Direct transmittance (D) matrix for azimuth model	3 0
Equation 18: Disturbance (E) matrix for azimuth system	3 1
Equation 19: States of altitude system	3 5
Equation 20: State equation of altitude system	3 7
Equation 21: Torque transformation of altitude model	3 8
Equation 22: Altitude Model State	3 8
Equation 23: Transform state equations for altitude model.....	3 9
Equation 24: State Space Model for Altitude	4 0
Equation 25: Altitude Model Input	4 0
Equation 26: Altitude Model Output	4 0
Equation 27: Star Image Displacement Calculation for altitude model	4 1
Equation 28: Reduced Star Image Displacement Equation of altitude model	4 1
Equation 29: Altitude Model Disturbance Source	4 1
Equation 30: System (A_a) matrix for altitude model	4 2
Equation 31: Input (B_a) matrix for altitude model	4 2
Equation 32: Output (C_a) matrix for altitude model.....	4 3
Equation 33: Direct transmittance (D_a) matrix for altitude model.....	4 3
Equation 34: Disturbance (E_a) matrix for altitude model.....	4 4

Chapter 1: Introduction

1.1: Background

The accuracy of optical telescopes in the past time was dominated by the construction of the telescope body and the precision of optical surface. In order to have an accurate optical telescope, the telescope must have an extremely rigid and heavy structure. The rigid telescope body can decrease structure deformation under external disturbances. A heavy telescope structure can help reduce vibration caused by impulse disturbance and telescope tracking motion. This kind of telescope (Shown in Figure 1) is stationary and constructed inside a shelter which makes it free from many environment disturbances. Moreover, the gear ratio in the drive chain is big in order to support and control heavy telescope structure; therefore the tracking motion is very slow.



Figure 1: Stationary Telescope - Hooker telescope at Mount Wilson Observatory¹

The FEA (Finite Element Analysis) model is a popular method for structure modeling for big telescopes. However, it is hard to design a control system using FEA model generated by a computer due to its complicated and usually oversized model. This kind of telescope control heavily relies upon empirical data. Each telescope is tested. Data Results are used to create a LUT (Lookup Table). The LUT includes the amount deformation occurs on the telescope structure under external disturbances. This LUT is be used as reference for control calibration.[1]

Some other telescope, because of its need to move and operate at different location, is constructed with a much lighter structure. An amateur

¹ Source: Photograph © Andrew Dunn, 1989.< <http://www.andrewdunnphoto.com/>>

telescope is an example of it (Shown in Figure 2). This kind telescope is required to be disassembled and moved to a remote location, usually outdoors. The operator will put it together at the location. The light structure telescope is susceptible to structure vibration and external disturbances. The first reason is that the maximum weight of the telescope is limited by the transportation tool and human power that put it together. Due to this weight limit, the construction of telescope isn't as stiff and heavy as stationary telescope. Second, a light structure telescope is usually operated in an environment with more external disturbances. Since there is no shelter or dome to protect it, disturbances like wind or a wild animal can directly affect the telescope.



Figure 2: Homemade telescope²

Besides being susceptible to vibration and deformation, it is also hard to calibrate a light structure telescope. Even though the telescope calibration

² Source: "amateur telescope making" < http://en.wikipedia.org/wiki/Amateur_telescope_making >

strategy using empirical data is common for a big stationary telescope, the same method is less useful for the light structure telescope. This is because of the uncontrollable environment and errors in the set up process. The telescope structure will be slightly different every single time. The calibration process is also time consuming and requires many precise measuring tools. It isn't feasible to perform it on a low price telescope every time an operator sets it up. On the other hand, modern telescopes have additional sensors. Many of these sensors could provide extra information about the telescope. Moreover, a high torque motor could be used as a direct drive motor on a light structure. The drive chain could be eliminated since the inertia of such a telescope is low. It offers a faster response and a more linear system. With the faster computer used today, it is possible to apply active control with additional feedback to improve performance of a light structure telescope.

1.2: Star Tracking and Telescope Control

1.2.1 Altazimuth Tracking

Many light structure telescopes are manual operated telescope. It requires the operator to manually point it at the target star. Since the star actually travels across the sky over night, the operator also needs to adjust its pointing so the target star will stay in the field of view. Some telescopes are computer controlled so it can track a single star over the sky in a long period of time. Two main

tracking strategies are used today.

1. Altazimuth Mounts
2. Equatorial Fork Mounts

Altazimuth method uses two motors with position control to control the azimuth and altitude of the telescope. The altitude and azimuth system is the coordination used to describe the pointing direction of telescope. Altitude and azimuth coordination is shown in Figure 3. In this figure, "h" represents altitude angular position. Altitude is measured as the angle from horizontal to the direction which the telescope pointing and it goes from 0 degrees to 180 degrees. "A" represents the azimuth angular position. Azimuth is measured as the angle from north of the direction of the telescope is pointing from a horizontal plane and it goes from 0 degrees to 360 degrees. These two values can fully define the telescope point in a half-sphere above the horizon.

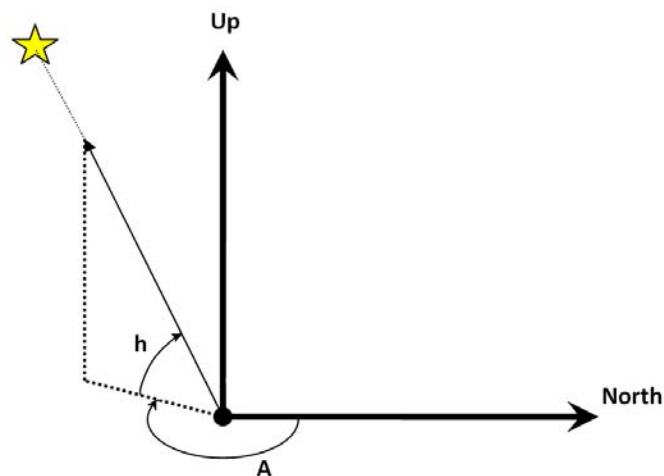


Figure 3: Altitude (h) and Azimuth (A) coordination

Altazimuth tracking is used for the telescope system discussed in this project since it is used by most of research telescopes and amateur telescopes today. Altazimuth tracking is more mechanically complicated compared to equatorial tracking. The equatorial tracking has a rotational axis parallel to the Earth's axis of rotation; hence it only needs to rotate about one single axis at a constant speed to follow a star in the sky. The Altazimuth tracking requires two axes, azimuth and altitude, to track a star over the sky. Two axes have to continually move at a different rate in order to track a celestial target. However, equatorial tracking requires a massive and precise mounting structure in order to make sure it is parallel to the Earth's axis of rotation. For an amateur, it is easier and cheaper to construct a two axes system than design a heavier and more expensive equatorial mount. The complex control issue can usually be solved by the integration of a microcontroller or computer controlled system.

1.2.2: Open Loop and Closed Loop Position Control

The open loop position control indicates that the telescope doesn't know where the star is [2]. It only knows which direction it is pointing. The open loop position control uses position sensors on the telescope rotation axis to measure the pointing angle. In the case of the Altazimuth mount, there is one encoder on each axis (Shown in Figure 4). These sensors are used to measure drive motor angular displacement. Then control system can calculate the altitude angle and azimuth angle of the telescope. Most of light structure telescope uses this

method to track a star.

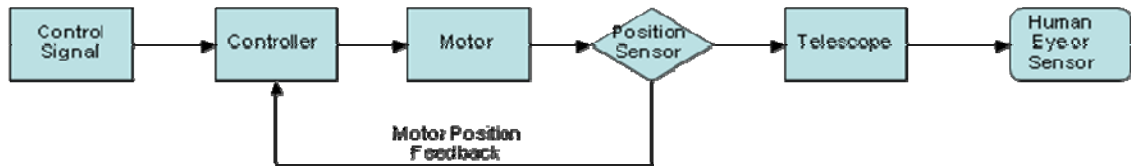


Figure 4: Telescope Open Loop Control flow chart

However, a telescope using an open loop control doesn't know if it is actually pointing at the target star. The vibration and deformation in the OTA (Optical Telescope Assembly) affects telescope pointing. They create errors that are not observable by the position sensors placed on the end of a drive chain. Though position sensors suggest that the telescope is pointing at the correct direction; the combined angular position, telescope pointing direction plus deformation of OTA, might not be correct. Most of stationary telescopes use empirical data to calibrate the control system. This method adds or subtracts a certain amount of an angle when the telescope is at a different location. For example: if the telescope is trying to aim at a star at a 24 degrees altitude, the LUT fixes it to 26 degrees. Two extra degrees is compensated for telescope deformation due to gravity. This method is also not feasible to a light structure telescope due to cost of time and precise tools. It also does not help compensate for external disturbance that is not repeatable.

Close loop position control uses the star as a feedback signal. The star sensor forms a second feedback loop (Shown in Figure 5). This method uses a star image sensor to measure the displacement of a reference star (a

bright, not-interfered star at known location is usually picked as reference star because it gives clearer location feedback) from where it is supposed to be. This information can be used for fine adjustment of telescope pointing. Since it is using an observed target as its feedback source, this method makes sure that the combined angular position is correct.

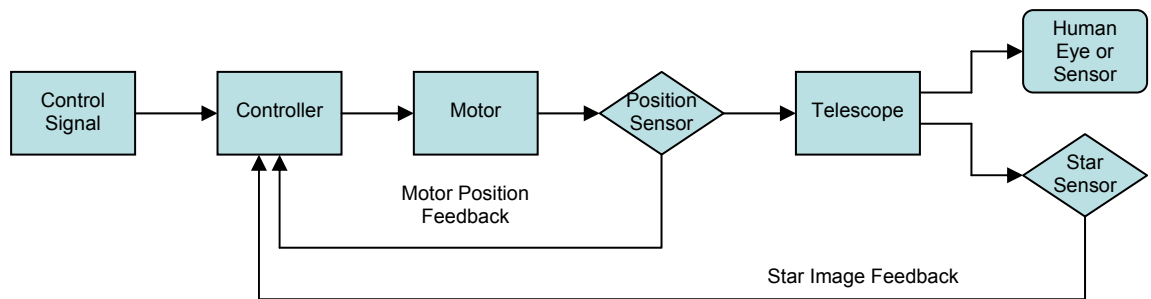


Figure 5: Telescope Closed Loop Control flow chart

1.2.3: Disturbances

Gravity produces repeatable deformation [2]. The effect of gravity on the telescope is due to the deformation of the telescope structure. The deformation is always in the vertical direction. OTA deformation due to gravitation is a function of the altitude angular position. If the altitude is small, the telescope will have a bigger deformation. Gravitational disturbances only exist in an altitude model. It won't affect the telescope motion on a horizontal plane.

Wind produces non-repeatable deformation [2]. The disturbance caused by wind has a slow and fast component. The slow component of wind disturbance is the steady wind. It applies a long and steady force on the telescope structure and causes the telescope to drift away from its target.

The steady wind is usually large in amplitude and could cause large telescope deformation. However, behavior of steady wind is usually more predictable. The fast component of the wind disturbance is wind gust. The wind gust is characterized by an abrupt and nearly discontinuous force on telescope body. However, the amplitude of wind gust is smaller and doesn't create as much deformation as a steady wind. The gust is usually 20% to 10% of steady wind [3]. The steady wind information can be monitored by an external weather station or observed using a feedback star image. The wind gust acts like noise from the environment. Due to gust's short time constant, it is hard to observe it from a weather station and is usually observed through a star feedback system.

The telescope model models the wind as a torque applied on the motor shaft. The amplitude of the torque caused by steady wind is a function of wind disturbance amplitude, wind direction, and the position of telescope at the moment. If the telescope structure is perpendicular to the wind direction, the effect of the wind is the biggest. The steady wind source is modeled by a square wave. The wind gust is white noise with amplitude equal to 20% to 10% of the square wave [3].

Like the wind, temperature also produces non-repeatable deformation. The change of temperature which affects telescope accuracy occurs on a rather long time; this is because the sudden change of air temperature usually has small amplitude and the telescope component usually has a large thermal time constant; therefore, the small temperature change usually won't change telescope geometry. The large temperature change which affects

telescope geometry occurs on a time scale of hours and days. Since the project focuses on the transient response, the telescope model does not count for the temperature in the environment.

Other disturbances include human contact, atmosphere air density change, ground vibration, and other light pollution. The accidental touching of the telescope structure can be modeled as a stronger wind gust. Other effects generally have less amplitude comparing with major disturbance and are ignored in the current model.

1.2.4: Stability Criteria

The response of telescope to control input and environmental disturbances changes the direction of telescope pointing. For a stable telescope, this change of pointing needs to remain in tolerance over a certain length of time. If the change of the angular pointing direction is too big, the star image will become blurring. The tolerance is determined by the observation target. The change of the angular pointing direction should not change by more than 1/10 of the beam's width. The equation is $\Delta\theta \leq \theta/10 \approx (\lambda/D)/10$ [2]. λ is wavelength of observation target and D is the main reflector diameter. The pointing stability is determined by the structure of the telescope.

1.3: Project Overview and Objective

Modern telescope is an integrated mechanical and electrical system. Its accuracy is challenged by telescope structure and external disturbances, as gravity, wind, and temperature change. Another issue is the internal friction in the telescope system. The internal friction is an issue for a light weight flexible telescope because this kind of telescope is not as well constructed as a VLT (very large telescope). It takes the telescope extra effort to overcome the static friction when it starts moving and usually causes an overshoot. It is hard to design a high performance control system for a light weight flexible telescope. Its light structure makes it susceptible to vibration and deformation. It is easily disturbed and usually operated under many environmental disturbances. Most importantly, many techniques used to calibrate research telescope can't be used on an individual telescope. In order to design a control system, it is important to gain a better understanding of the telescope system. Many of light structure telescopes are custom-build and modified for the user. Each of them has distinct structure characteristics, and therefore a specific model does not help too much.

This project describes a light structure flexible telescope as a generalized flexible telescope structure. It fits into a larger project with a goal of controlling a flexible telescope in order to help a cheaper telescope achieve higher accuracy. The model of a flexible telescope provides a telescope dynamic in state-space format since state space model is better for modeling and analyzing a multiple inputs and multiple outputs system. The model includes a transient flexible

response in order to represent the telescope response to a control input or a disturbance. It models the telescope tracking motion but also the vibration of each vibrating elements. In this case, it is important that the model can demonstrate the first vibration mode and simulates pointing error of the telescope since the first vibration mode causes the greatest pointing error of the telescope tracking. This model also models environment disturbance. Gravity and wind is considered in the model. The model concentrates on modeling the transient response of telescope. The effect of temperature change is not considered since its time constant is a lot longer.

The goal of this project is to construct a model for a light weight flexible telescope serving as a platform for testing active control system and system identification. In order to exam that if the telescope simulation model can represent a real telescope system, a one degree freedom experimental test stand is built to generate a series of telescope motion data. The simulation model is not a dynamic model for this experimental test stand; however, the model should be able to reproduce many system patterns shown in the experimental data since the general shape of simulation model and experimental is the same. The simulation of an experimental test stand is analyzed to determine the controllability and observability of the system. It shows the minimum requirement of sensor feedbacks in order to enable the operator to capture all the major and minor motion during the telescope motion.

Chapter 2: Model of Flexible Telescope

This chapter presents the modeling of the light weight flexible telescope mechanical and electrical system. The telescope structure consists of azimuth motor, altitude motor, telescope fork, OTA (Optical Telescope Assembly), mirror(main reflector), and mirror mount. The telescope electronics includes two driving motors. This model also counts control inputs used to perform telescope tracking and possible external disturbances that affect telescope performance.

2.1: Telescope System

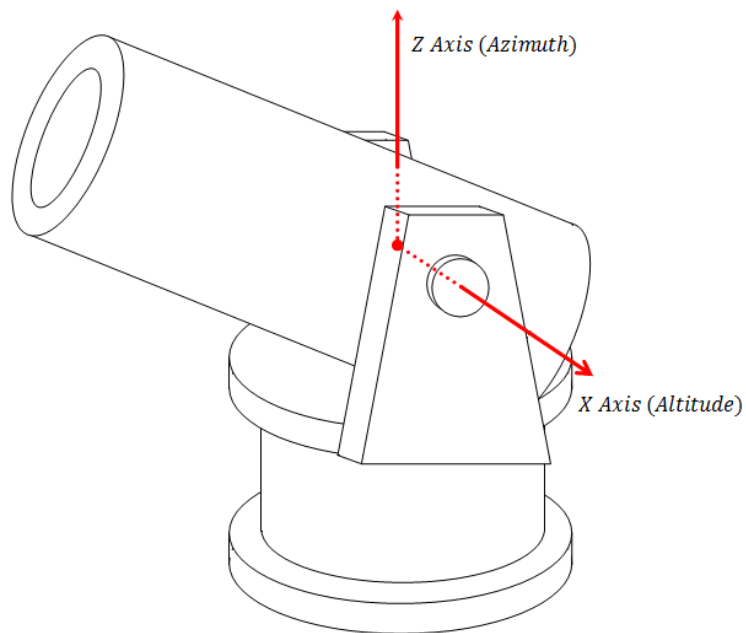


Figure 6: Generalized Telescope System

To model the motion of a flexible telescope, a general shape

telescope with Altazimuth tracking system is considered (shown in Figure 6). The cylindrical body on the top is the body of telescope. It contains the OTA, the main reflector (mirror), and the mirror mount. The bottom part is the telescope mount. It contains the fork, two driving motor, and all the controller components. The generalized telescope model, without flexibility, is a system with two degree of freedom. The telescope rotates around two axes. Azimuth motion rotates around Z-axis and altitude motion rotates around X-axis. Two axes intersect at center of gravity of telescope body; therefore, without disturbance, the telescope should be a stable system which can stay still at any position.

2.1.1: Structure Components

The generalized telescope mechanical system has five major components (Shown in Figure 7 and Table 1). Red dot on Figure 7 is the center of gravity and the rotational center of the telescope body. Mirror mount is the rear end of the telescope body which holds telescope mirror. OTA extends out from the mirror mount. The observing point is at the opposite end of the OTA, illustrated with a green dot on the Figure 7. The fork is the entire supporting mount of the telescope. When models a telescope mechanical system, two motor and drive chain is part of the fork. The fork, OTA and the mirror mount together is the telescope structure.

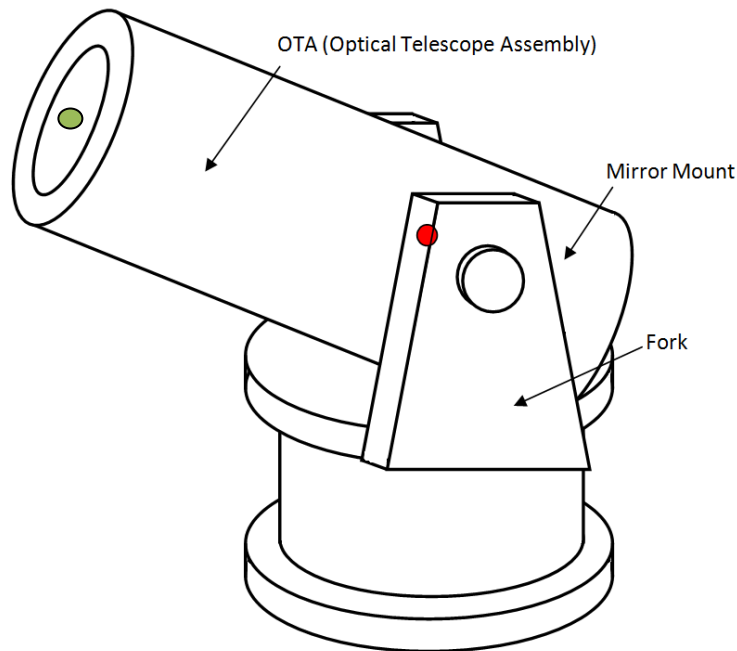


Figure 7: Telescope System Components

Two motors are the electrical-mechanical system. They convert electric current in the electrical system into torque. Azimuth motor and altitude motor provide torque to two axes of telescope mechanical system. The generalized telescope uses direct drive motor; therefore, there is no gear chain between telescope axes and motor. They change angle between telescope body and telescope mount. These two motor receive control input from the position controller.

Table 1: Telescope system components

Parts Name	Description
Azimuth Motor	Motor that controls azimuth motion of telescope
Altitude Motor	Motor that controls altitude motion of telescope
Fork	The stand of telescope; support OTA and lift it from the ground
OTA	Front part of telescope body, from where fork contacts with body to where image receiver is
Mirror and Mirror Mount	Rear part of telescope body, from where fork contacts with body to where primary mirror is

2.1.2: Control Inputs

Control inputs are part of the electric system of the telescope. The generalized telescope control system has two inputs (shown in Table 2). Telescope control inputs are generated by the position controller. These two voltage signals are sent to the power amplifier. The amplified signals are sent to the motor and used to control motor output.

Table 2: Control Inputs of telescope

Input Type	Description
Azimuth Motor	Voltage input, controls azimuth motion
Altitude Motor	Voltage input, controls altitude motion

2.1.3: Sensor Feedback

Telescope systems have three different sensor feedbacks (shown in Table 3). First two sensors are azimuth encoder and altitude encoder. In a generalized telescope model, they are considered as ideal position feedback sensors which give motor angular position in radius with an infinite small resolution. They are directly connected to the motor. Since there is not gear chain in the mechanical system, the outputs are also an angular position of general telescope pointing direction. The feedbacks from encoders are illustrated by an arrow between position sensor and position controller.

The star image sensor is mounted on the extended end of OTA. Its location is illustrated with the green dot in Figure 7. In the generalized telescope model, the output of star image sensor is the displacement between desired

telescope pointing center and combined pointing center, which also counts for the fork, OTA, and mirror mount vibration. It produces a 2 degrees displacement on XY coordination. The feedback from the star image sensor is illustrated by arrow between the reflectors/receivers and the position controller. In the generalized model, the sensor images a reference star that is always at the actual center of telescope pointing; therefore, always provides a feedback. In a real system, the star image feedback only works when the reference star is in the field of view, which means the telescope open loop controller has to be able to bring the telescope pointing direction close to the reference star location before the star image sensor starts to work.

Table 3: Sensor Feedback of telescope

Feedback Type	Description
Azimuth encoder	Gives angular position and angular velocity in azimuth direction
Altitude encoder	Gives angular position and angular velocity in altitude
Star image	Gives target star location in XY coordination

2.2: Model of External Disturbances

2.2.1: Gravity

The gravitational disturbance is modeled as torque applied on the altitude model. Gravitational torque is a function of telescope length, telescope pose, and the mass of instrument mounted on the end of upper OTA and mirror mount. The calculation is shown in Equation 1.

Equation 1: Gravitational disturbance

$$T_{Gravity} = -m_{instrument} \times g \times \cos(\theta_{axis}) \times L_{telescope}$$

The angular position change of the telescope in the altitude system is from 0 to 90 degrees; therefore, the torque is nonlinear. The torque is calculated externally from the system model and brought in the model as input. Gravitational disturbances caused by the OTA and the mirror mount should balance each other since the telescope is a balanced system. However, gravitational force still bends the OTA and the mirror mount slightly hence create a star image displacement.

2.2.2: Wind

The wind disturbance is modeled as torque applied on both the altitude and azimuth model. The pressure from the wind force is $P_{wind} = 0.5\rho_{air}V_w^2C$ [6]. V_w is the speed of wind and C is the shape factor of the telescope body. C is assumed to be 1 in this project since the true shape of telescope is unknown. From wind pressure, the torque applied on rotational center of telescope is calculated by multiplying the wind pressure by the telescope section area and half of the length of the telescope (shown in Equation 2).

Equation 2: Wind disturbance

$$T_{wind} = 0.5 \times \rho_{air} \times [V_{wind} \times \cos(\theta_j - \theta_{wind}) \times \sin\theta_j]^2 \times Area_{telescope} \times 0.5 \times L_{telescope}$$

The angular position change of the telescope is 0 to 90 degree in the

altitude system and 0 to 360 degree in the azimuth system; therefore, the torque is nonlinear. The torque is calculated externally from the system model and brought in the model as input.

2.3: Azimuth System

Generalized telescope model is separated into two parts: azimuth model and altitude model. The azimuth and the altitude system is treated as uncoupled in current model; however, in a real telescope there are some coupling between those axes. One of the coupling sources is the gyroscopic effect. Nevertheless, the telescope is usually operated under low speed with very low rotational angular acceleration; therefore, coupling between two subsystems can be ignored. The azimuth model describes telescope motion around Z axis, and the altitude model describes telescope motion around X axis.

2.3.1: Model of Azimuth System

Structure components of azimuth model include azimuth motor, OTA, mirror mount, and Fork. In the azimuth model, the altitude model is considered part of the fork. The relationship between each structure component is shown in Figure 8. The springs and dampers shown in the figure are torsional springs and torsional dampers. The azimuth motor is connected to the ground which provides absolute zero reference for the angular velocity of mechanical system. The azimuth system has one control input. When input voltage is sent to azimuth

motor, rotor A will gain velocity while velocity of rotor B will be zero. The angle between rotor A and rotor B is Θ_J . A rotational spring and damper connects rotor A and the fork/altitude motor. The angular position of the fork is Θ_{Jam} . This angular position is the absolute angular position, the angle between the fork and ground zero (North). The fork/altitude is connected to the mirror mount and OTA with two separated rotational springs and damper sets. When the fork/altitude motor displaces, torque will be sent to the mirror mount and OTA through the spring and damper set; therefore, moving mirror mount and OTA. The entire system will move with the input from the azimuth motor. The angular position of OTA is Θ_O and angular position of mirror mount is Θ_m . The detail parameters for azimuth model are shown in Table 4)

Table 4: Modeling Parameters for Azimuth Model

Azimuth Model		
Parameters	Description	Unit
J	Inertia of azimuth control motor	$kg \cdot m^2$
J_o	Inertia of OTA	$kg \cdot m^2$
J_m	Inertia of mirror mount	$kg \cdot m^2$
J_f	Inertia of fork and altitude motor	$kg \cdot m^2$
K_o	Spring of OTA around z axis in horizontal direction	$N \cdot m / radius$
K_m	Spring of mirror mount around z axis in horizontal direction	$N \cdot m / radius$
K_f	Spring of fork around z axis in horizontal direction	$N \cdot m / radius$
B	Damping coefficient of azimuth control motor	$N \cdot m \cdot s / radius$
B_o	Damping coefficient of OTA in horizontal direction	$N \cdot m \cdot s / radius$
B_m	Damping coefficient of mirror mount in horizontal direction	$N \cdot m \cdot s / radius$
B_f	Damping coefficient of fork around z axis	$N \cdot m \cdot s / radius$
R	Internal resistance of azimuth motor system	Ω
L	Internal inductance of azimuth motor system	H
K_a	Motor constant of azimuth motor system	$N \cdot m / Amps$
States	Description	Unit
θ_J	Angular position of azimuth pointing	<i>Radius</i>
θ_{Jo}	Angular position of OTA	<i>Radius</i>
θ_{Jm}	Angular position of mirror mount	<i>Radius</i>
θ_{Jf}	Rotation angle of fork and altitude motor around z axis	<i>Radius</i>
Ω_J	Azimuth angular velocity of azimuth pointing	<i>Radius / s</i>
Ω_{Jo}	Azimuth angular velocity of OTA	<i>Radius / s</i>
Ω_{Jm}	Azimuth angular velocity of mirror mount	<i>Radius / s</i>
Ω_{Jf}	Azimuth angular velocity of fork and altitude motor	<i>Radius / s</i>
i_L	Current through azimuth control motor	<i>Amps</i>
Source	Description	Unit
V_s	Input voltage for azimuth control motor	<i>V</i>
Disturbance	Description	Unit
T_{s1}	Wind disturbance act on OTA for azimuth model	$N \cdot m$
T_{s2}	Wind disturbance act on mirror mount for azimuth model	$N \cdot m$
Output	Description	Unit
θ_J	Angular position of azimuth as output	<i>Radius</i>
Ω_J	Azimuth angular velocity	<i>Radius</i>
X_t	Guide star feedback sensor for azimuth coordination	<i>Radius</i>

2.3.2: Azimuth Model State Equations

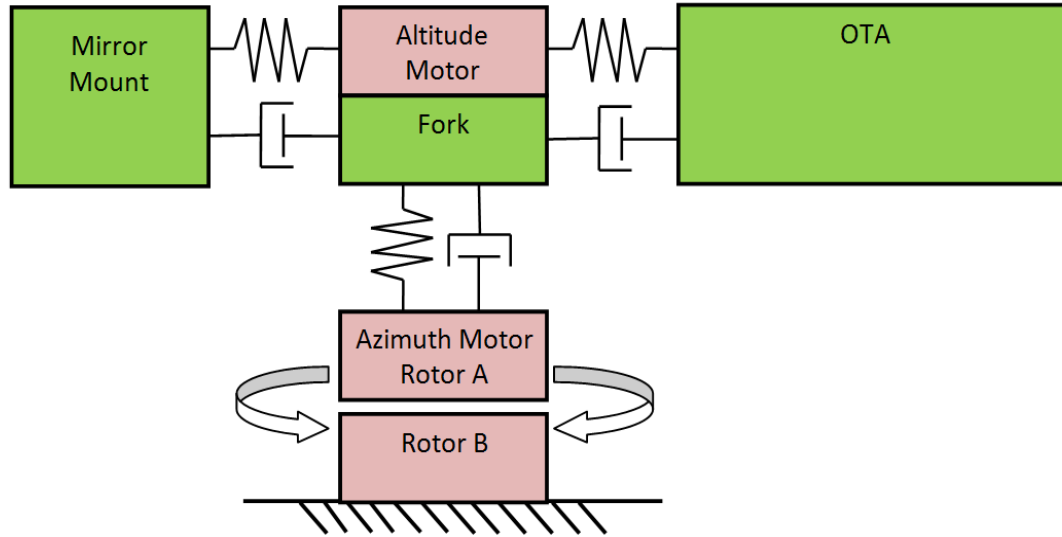


Figure 8: Azimuth Telescope Model

Graphical technique is used to represent the dynamics system model [5]. The linear graph represents the topological relationships of lumped-element interconnections within system [5]. Linear graph of azimuth model is shown in Figure 9. This graph is used represents both the electrical and mechanical system. The loop on the left is the model of DC motor. The bigger loop on the right is the model of the telescope mechanical system. The component connecting the two systems is electric mechanical component motor. The motor serve as a transformer in between two loops transfer current (shown as 1 on Figure 9) to torque (shown as 2 on Figure 9). From the linear graph, the system has 8 states (Shown in Equation 3).

Equation 3: States of azimuth system

$$X = [\Omega_J, \Omega_{Jam}, \Omega_{Jo}, \Omega_{Jm}, T_{Kf}, T_{Ko}, T_{Km}, i_L]$$

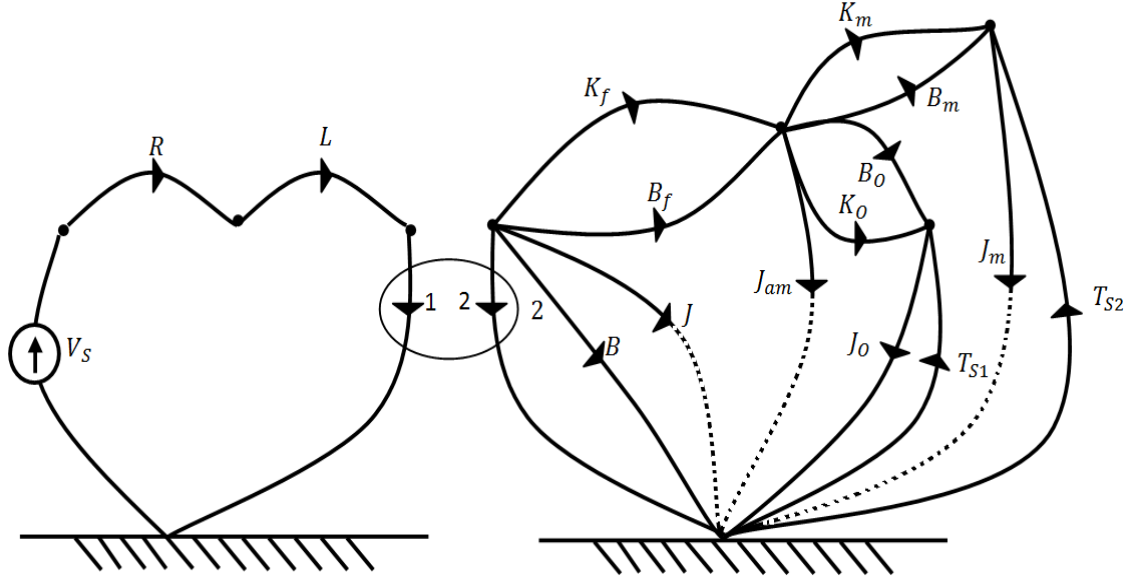


Figure 9: Azimuth Telescope Model Tree Diagram

Eight state variables are used to generate 8 state equations to represent dynamic motion in the azimuth system (Shown in Equation 4). The detail derivation is shown in Appendix A.

Equation 4: State space equation of azimuth model

$$\begin{aligned}
 \frac{d\Omega_J}{dt} &= \frac{1}{J} \left[\frac{1}{K_a} i_L - B\Omega_J - T_{Kf} - B_f(\Omega_J - \Omega_{Jam}) \right] \\
 \frac{d\Omega_{Jam}}{dt} &= \frac{1}{J_{am}} [T_{Kf} + B_f(\Omega_J - \Omega_{Jam}) - T_{Ko} - B_o(\Omega_{Jam} - \Omega_{Jo}) - T_{Km} - B_m(\Omega_{Jam} - \Omega_{Jm})] \\
 \frac{d\Omega_{Jo}}{dt} &= \frac{1}{J_o} [T_{Ko} + B_o(\Omega_{Jam} - \Omega_{Jo}) + T_{s1}] \\
 \frac{d\Omega_{Jm}}{dt} &= \frac{1}{J_m} [T_{Km} + B_m(\Omega_{Jam} - \Omega_{Jm}) + T_{s2}] \\
 \frac{di_L}{dt} &= \frac{1}{L} (V_s - Ri_L - \frac{1}{Ka} \Omega_J) \\
 \frac{dT_{Kf}}{dt} &= \frac{1}{J_{am}} (\Omega_J - \Omega_{Jam}) \\
 \frac{dT_{Ko}}{dt} &= \frac{1}{J_o} (\Omega_{Jam} - \Omega_{Jo}) \\
 \frac{dT_{Km}}{dt} &= \frac{1}{J_m} (\Omega_{Jam} - \Omega_{Jm})
 \end{aligned}$$

Four states T_{Kf}, T_{Ko}, T_{Km} shown in Equation 3 are hard to be measured in the azimuth system with accessible sensors. However, it is easier to transform them into a different format. Torque between two components can be rewritten as angular displacement multiplied by the rotational spring constant (Shown in Equation 5).

Equation 5: Torque transformation of azimuth model

$$\begin{aligned}
 T_{Kf} &= K_f (\theta_J - \theta_{Jam}) \\
 T_{Ko} &= K_o (\theta_{Jam} - \theta_o) \\
 T_{Km} &= K_m (\theta_{Jam} - \theta_m)
 \end{aligned}$$

Azimuth model states have now become angular positions and angular

velocities of system components and current in the electric system. New azimuth models states is shown in Equation 6. With the new state variables, state equation is transformed also. New state equation is shown in Equation 7.

Equation 6: Transformed azimuth model States

$$X = \begin{bmatrix} \theta_J \\ \theta_{Jam} \\ \theta_{JO} \\ \theta_{Jm} \\ \Omega_J \\ \Omega_{Jf} \\ \Omega_{JO} \\ \Omega_{Jm} \\ i_L \end{bmatrix}_{9 \times 1}$$

Equation 7: Transformed state equation for azimuth model

$$\begin{aligned}\frac{d\theta_J}{dt} &= \Omega_J \\ \frac{d\theta_{Jam}}{dt} &= \Omega_{Jam} \\ \frac{d\theta_{Jo}}{dt} &= \Omega_{Jo} \\ \frac{d\theta_{Jm}}{dt} &= \Omega_{Jm} \\ \frac{d\Omega_J}{dt} &= \frac{1}{J} \left[\frac{1}{K_a} i_L - B\Omega_J - K_f(\theta_J - \theta_{Jam}) - B_f(\Omega_J - \Omega_{Jam}) \right] \\ \frac{d\Omega_{Jam}}{dt} &= \frac{1}{J_{am}} [K_f(\theta_J - \theta_{Jam}) + B_f(\Omega_J - \Omega_{Jam}) - K_o(\theta_{Jam} - \theta_{Jo}) - B_o(\Omega_{Jam} - \Omega_{Jo}) - K_m(\theta_{Jam} - \theta_{Jm}) \\ &\quad - B_m(\Omega_{Jam} - \Omega_{Jm})] \\ \frac{d\Omega_{Jo}}{dt} &= \frac{1}{J_o} [K_o(\theta_{Jam} - \theta_{Jo}) + B_o(\Omega_{Jam} - \Omega_{Jo}) + T_{s1}] \\ \frac{d\Omega_{Jm}}{dt} &= \frac{1}{J_m} [K_m(\theta_{Jam} - \theta_{Jm}) + B_m(\Omega_{Jam} - \Omega_{Jm}) + T_{s2}] \\ \frac{di_L}{dt} &= \frac{1}{L} (V_s - Ri_L - \frac{1}{Ka}\Omega_J)\end{aligned}$$

2.3.3: State-space Model of Azimuth System

8 states variables are the minimum set of variables which can be used to describe azimuth system. System outputs are system variables that are accessible and can be measured by the sensors (shown in Equation 10). There are 3 system outputs we can obtain from sensors. They can generate 3 output equations. 8 state variables are substituted with 9 state variables. These 9 state equations and output equations are rewritten in state-space format (shown in Equation 8).

Equation 8: State Space Model for Azimuth

$$\begin{aligned}\dot{X} &= A_{9 \times 9} X_{9 \times 1} + B_{9 \times 1} U_{1 \times 1} + E_{9 \times 2} d_{2 \times 1} \\ \dot{Y} &= C_{3 \times 9} X_{9 \times 1} + D_{3 \times 1} U_{1 \times 1}\end{aligned}$$

The system input of azimuth model is voltage sent to the electric system (Shown in Equation 9)

Equation 9: Azimuth Model Input

$$U = [V_s]_{1 \times 1}$$

There are 3 system outputs. Angular position and angular velocity of the motor could be obtained from azimuth encoder. X_t is the output from the star image sensor. It is the displacement of the reference star.

Equation 10: Azimuth Model Output

$$Y = \begin{bmatrix} \theta_j \\ \Omega_j \\ X_t \end{bmatrix}_{3 \times 1}$$

X_t is a function of mirror mount angular position and OTA angular position of the azimuth system (shown in Figure 10). Angular displacements of the OTA and the mirror mount both create image displacements. The star image displacement is a combination of these two displacements. This relationship is listed in Equation 11. Using small angle approximation, Equation 11 can be linearized into a simpler format (shown in Equation 12). Notice the term θ_{Jam} is canceled in the equation. Vibration of fork does not affect azimuth pointing.

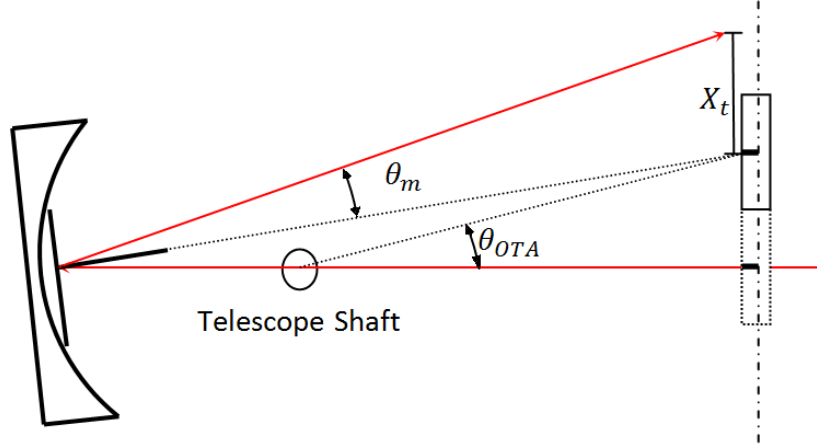


Figure 10: Star Image Displacement

Equation 11: Star Image Displacement Calculation

$$X_t = L_{OTA} \times \sin\{2[(\theta_{Jm} - \theta_{Jam}) + (\theta_{Jam} - \theta_J)]\} - L_{OTA} \times \sin[(\theta_{Jo} - \theta_{Jam}) + (\theta_{Jam} - \theta_J)]$$

Equation 12: Reduced Star Image Displacement Equation

$$X_t = -L_{OTA}\theta_J - L_{OTA}\theta_{Jo} + 2L_{OTA}\theta_{Jm}$$

There are two disturbance sources in the azimuth model. The wind torque applied on OTA and the wind torque applied on mirror mount. In azimuth motion, wind effect on fork is small; therefore, it is ignored.

Equation 13: Azimuth Model disturbance Source

$$d = \begin{bmatrix} T_{s1} \\ T_{s2} \end{bmatrix}_{2 \times 1}$$

All system matrices can be determined by transformer 9 state equations and 3 output equations. System matrices are shown below. A matrix shown in Equation 14 is the state matrix of the azimuth system.

Equation 14: State (A) matrix for azimuth model

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{(K_o + K_m)}{J} & 0 & \frac{K_o}{J} & \frac{K_m}{J} & -\frac{(B + B_o + B_m)}{J} & \frac{B}{J} & \frac{B_o}{J} & \frac{B_m}{J} & \frac{1}{K_a J} \\ 0 & -\frac{K_f}{J_f} & 0 & 0 & \frac{B}{J_f} & -\frac{(B + B_f)}{J_f} & 0 & 0 & -\frac{1}{K_a J_f} \\ \frac{K_o}{J_o} & 0 & -\frac{K_o}{J_o} & 0 & \frac{B_o}{J_o} & 0 & -\frac{B_o}{J_o} & 0 & 0 \\ \frac{K_m}{J_m} & 0 & 0 & -\frac{K_m}{J_m} & \frac{B_m}{J_m} & 0 & 0 & -\frac{B_m}{J_m} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{K_a L} & \frac{1}{K_a L} & 0 & 0 & -\frac{R}{L} \end{bmatrix}_{9 \times 9}$$

B matrix shown in Equation 15 is the input matrix of the azimuth system.

Equation 15: Input (B) matrix for azimuth model

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}_{9 \times 1}$$

C matrix shown in Equation 16 is the output matrix of the azimuth system.

Output variables θ_j and Ω_j is a direction output from state variables. X_j is calculated using Equation 12.

Equation 16: Output (C) matrix for azimuth model

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{3 \times 9}$$

D matrix shown in Equation 17 is the direct transmittance matrix for the azimuth model. In the azimuth system, input of the system doesn't affect outputs; therefore transmittance matrix is a zero matrix.

Equation 17: Direct transmittance (D) matrix for azimuth model

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{3 \times 1}$$

E matrix shown in Equation 18 is the disturbance matrix for the azimuth system.

Equation 18: Disturbance (E) matrix for azimuth system

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{J_o} & 0 \\ 0 & \frac{1}{J_m} \\ 0 & 0 \end{bmatrix}_{9 \times 2}$$

2.4: Altitude System

Altitude model describes the telescope motion around the X axis.

2.4.1: Model of Altitude System

The structure of the altitude model is similar to the azimuth model. Components include altitude motor, OTA, mirror mount, and Fork. In the altitude model, the azimuth model is considered part of the ground. The relationship between each structure component is shown in Figure 11. The springs and dampers shown in the figure are torsional springs and torsional dampers. In the altitude model, the fork is connected to the ground which provides absolute zero reference for the angular velocity of mechanical system. The altitude motor sits on the top of the fork and provide torque source to all 3 other components. The altitude system has one control input. When input voltage is sent to the altitude

motor, both rotor A, which is connected to mirror mount and OTA, and rotor B, which is connected to the fork will gain velocity. The angle between rotor A and rotor B of altitude motor is θ_{Ja} . Rotor B is connected to the fork and has the same angular displacement as fork Θ_f . Since the fork is connected to the ground, the angular displacement of rotor B is limited by the spring element of fork. The altitude motor is connected to the mirror mount and OTA with two separated rotational spring and damper sets. When altitude motor displaces, torque will be sent to mirror mount and OTA through the spring and damper set; therefore, moving mirror mount and OTA. The OTA and mirror mount move with rotor A of the altitude motor. The angular position of fork oscillates and goes to zero. The angular position of OTA is θ_{O_a} and angular position of mirror mount is θ_{m_a} . The detail parameters for azimuth model are shown in Table 5 and table 6.

Table 5: Modeling parameters for Altitude Model (Part1)

Altitude Model		
Parameters	Description	Unit
$J_{_a}$	Inertia of altitude control motor	$kg \cdot m^2$
J_{O_a}	Inertia of OTA	$kg \cdot m^2$
J_{m_a}	Inertia of mirror mount	$kg \cdot m^2$
J_{f_a}	Inertia of fork	$kg \cdot m^2$
K_{O_a}	Spring of OTA around z axis in vertical direction	$N \cdot m / radius$
K_{m_a}	Spring of mirror mount around z axis in vertical direction	$N \cdot m / radius$
K_{f_a}	Spring of fork around z axis in vertical direction	$N \cdot m / radius$
$B_{_a}$	Damping coefficient of altitude control motor	$N \cdot m \cdot s / radius$
B_{O_a}	Damping coefficient of OTA in vertical direction	$N \cdot m \cdot s / radius$
B_{m_a}	Damping coefficient of mirror mount in vertical direction	$N \cdot m \cdot s / radius$
B_{f_a}	Damping coefficient of fork around y axis	$N \cdot m \cdot s / radius$
$R_{_a}$	Internal resistance of altitude motor system	Ω
$L_{_a}$	Internal inductance of altitude motor system	H
K_{a_a}	Motor constant of altitude motor system	$N \cdot m / Amps$
States	Description	Unit
θ_{J_a}	Angular position of altitude point	<i>Radius</i>
θ_{JO_a}	Altitude angular position of OTA	<i>Radius</i>
θ_{Jm_a}	Altitude angular position of mirror mount	<i>Radius</i>
θ_{Jf_a}	Altitude angular position of fork around z axis	<i>Radius</i>
Ω_{J_a}	Altitude angular velocity of altitude point	<i>Radius / s</i>
Ω_{JO_a}	Altitude angular velocity of OTA	<i>Radius / s</i>
Ω_{Jm_a}	Altitude angular velocity of mirror mount	<i>Radius / s</i>
Ω_{Jf_a}	Altitude angular velocity of fork	<i>Radius / s</i>
i_{L_a}	Current through altitude control motor	<i>Amps</i>
Source	Description	Unit
V_{s_a}	Input voltage for altitude control motor	<i>V</i>
Disturbance	Description	Unit
T_{s1_a}	Wind disturbance act on OTA for altitude model	$N \cdot m$
T_{s2_a}	Wind disturbance act on mirror mount for altitude model	$N \cdot m$
T_{s3_a}	Wind disturbance act on fork	$N \cdot m$
T_{s4_a}	Torque from gravitational force on OTA	$N \cdot m$
T_{s5_a}	Torque from gravitational force on mirror mount	$N \cdot m$

Table 6: Modeling parameters for Altitude Model (Part2)

Altitude Model		
Output	Description	Unit
θ_{J_a}	Angular position of altitude as output	<i>Radius</i>
Ω_{J_a}	Altitude angular velocity	<i>Radius</i>
Y_{t_a}	Guide star feedback sensor for altitude	<i>Radius</i>

2.4.2: Altitude Model State Equations

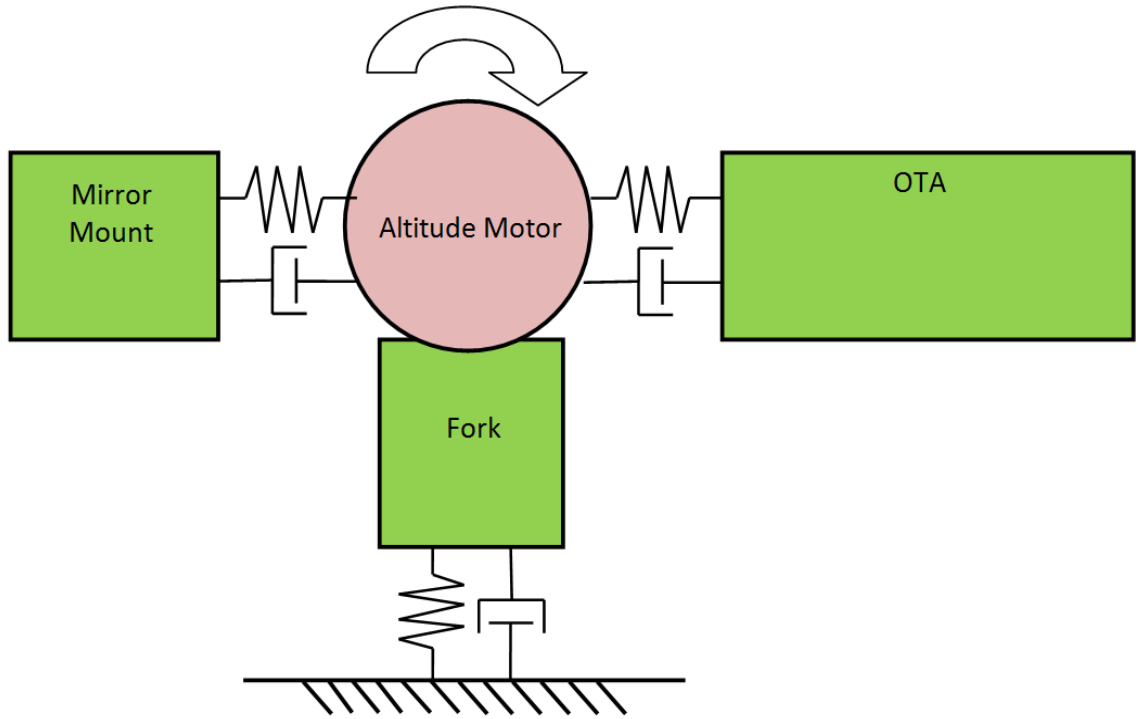


Figure 11: Altitude Telescope Model

Graphical technique is used to represent dynamics system model [5]. Linear graph represent the topological relationships of lumped-element interconnections within system [5]. Linear graph of altitude model is shown

in Figure 12. This graph is used represents both the electrical and mechanical system. The loop on the left is the model of the DC motor. This part is identical to the azimuth system. The bigger loop on the right is the model of the telescope mechanical system. It is different with azimuth due to its different structure. The component connecting the two systems is electric mechanical component motor. The motor serve as a transformer in between two loops transfer current (shown as 1 on Figure 12) to torque (shown as 2 on Figure 12). From the linear graph, the system has 8 states (Shown in Equation 3). These 8 states are used to generate 8 state equations to represent dynamic motion in the azimuth system

Equation 19: States of altitude system

$$X = \left[\Omega_{J_a}, \Omega_{Jf_a}, \Omega_{Jo_a}, \Omega_{Jm_a}, T_{Kf_a}, T_{Ko_a}, T_{Km_a}, i_{L_a} \right]$$

Eight state variables are used to generate 8 state equations to represent dynamic motion in the altitude system (Shown in Equation 20). The detail derivation is shown in Appendix B.

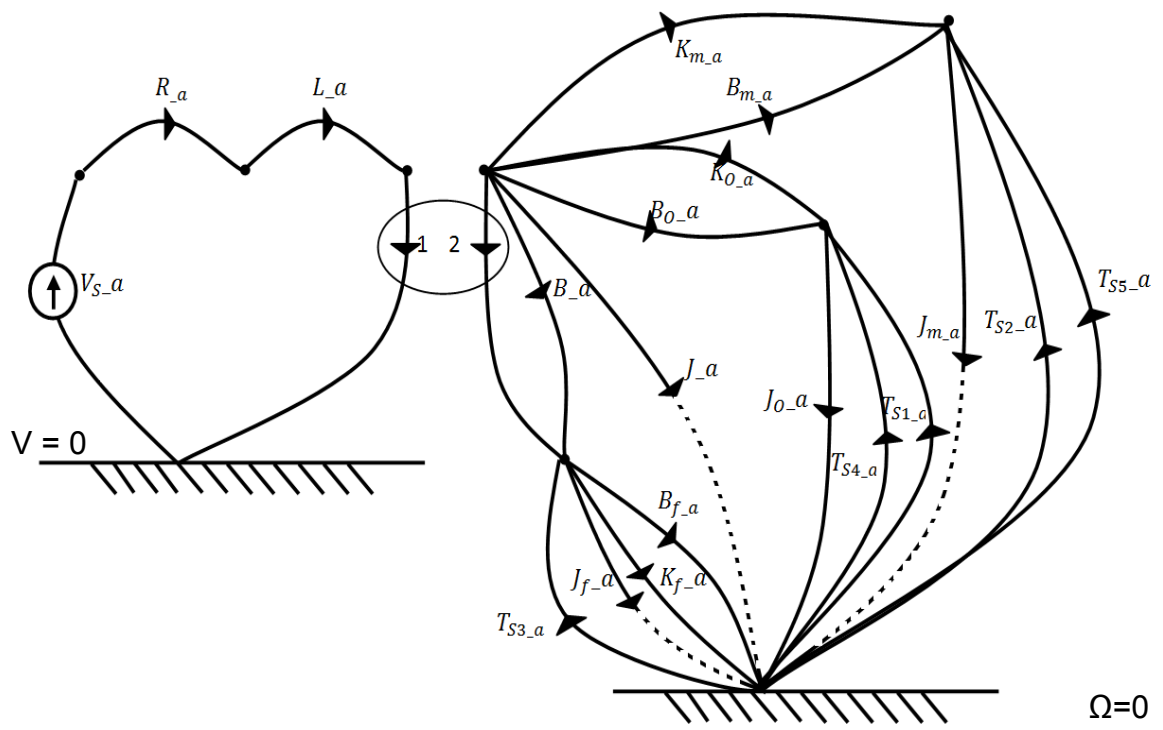


Figure 12: Altitude Telescope Model Tree Diagram

Equation 20: State equation of altitude system

$$\frac{d\Omega_{J_a}}{dt} = \frac{1}{J_{_a}} \left[\frac{1}{K_{a_a}} i_{L_a} - B_{_a} (\Omega_{J_a} - \Omega_{Jf_a}) - T_{Ko_a} - B_{o_a} (\Omega_{J_a} - \Omega_{Jo_a}) - T_{Km_a} - B_{m_a} (\Omega_{J_a} - \Omega_{Jm_a}) \right]$$

$$\frac{d\Omega_{Jf_a}}{dt} = \frac{1}{J_{f_a}} \left[\frac{-1}{K_{a_a}} i_{L_a} + B_{_a} (\Omega_{J_a} - \Omega_{Jf_a}) - T_{Kf_a} - B_{f_a} \Omega_{Jf} + T_{s3} \right]$$

$$\frac{d\Omega_{Jo_a}}{dt} = \frac{1}{J_{o_a}} [T_{Ko_a} + B_{o_a} (\Omega_{J_a} - \Omega_{Jo_a}) + T_{s1_a} + T_{s4_a}]$$

$$\frac{d\Omega_{Jm_a}}{dt} = \frac{1}{J_{m_a}} [T_{Km} + B_{m_a} (\Omega_{J_a} - \Omega_{Jm_a}) + T_{s2_a} + T_{s5_a}]$$

$$\frac{di_{L_a}}{dt} = \frac{1}{L_{_a}} (V_s - Ri_L - \frac{1}{Ka} (\Omega_{J_a} - \Omega_{Jf_a}))$$

$$\frac{dT_{Kf_a}}{dt} = \frac{1}{J_{f_a}} \Omega_{Jf_a}$$

$$\frac{dT_{Ko_a}}{dt} = \frac{1}{J_{o_a}} (\Omega_{J_a} - \Omega_{Jo_a})$$

$$\frac{dT_{Km_a}}{dt} = \frac{1}{J_{m_a}} (\Omega_{J_a} - \Omega_{Jm_a})$$

Four states $T_{Kf_a}, T_{Ko_a}, T_{Km_a}$ shown in Equation 19 are hard to be measured in the altitude system with accessible sensors. However, it is easier to transform them into a different format. Torque between two components can be rewritten as angular displacement multiplied by the rotational spring constant (Shown in Equation 21). θ_f defines angular displacement of fork; therefore, torque in fork is only a function of θ_f and K_f .

Equation 21: Torque transformation of altitude model

$$\begin{aligned}
T_{Kf_a} &= K_{f_a} \theta_{f_a} \\
T_{Ko_a} &= K_{O_a} (\theta_{J_a} - \theta_{O_a}) \\
T_{Km_a} &= K_{m_a} (\theta_{J_a} - \theta_{m_a})
\end{aligned}$$

Altitude model states have now become angular positions and angular velocities of system components and current in the electric system. Altitude models states is shown in Equation 22. With the new state variables, state equation is transformed also. New state equation is shown in Equation 23.

Equation 22: Altitude Model State

$$X = \begin{bmatrix} \theta_{J_a} \\ \theta_{Jf_a} \\ \theta_{JO_a} \\ \theta_{Jm_a} \\ \Omega_{J_a} \\ \Omega_{Jf_a} \\ \Omega_{JO_a} \\ \Omega_{Jm_a} \\ i_{L_a} \end{bmatrix}_{9 \times 1}$$

Equation 23: Transform state equations for altitude model

$$\begin{aligned}
\frac{d\theta_{J_a}}{dt} &= \Omega_{J_a} \\
\frac{d\theta_{Jf_a}}{dt} &= \Omega_{Jf_a} \\
\frac{d\theta_{Jo_a}}{dt} &= \Omega_{Jo_a} \\
\frac{d\theta_{Jm_a}}{dt} &= \Omega_{Jm_a} \\
\frac{d\Omega_{J_a}}{dt} &= \frac{1}{J_{_a}} \left[\frac{1}{K_{a_a}} i_{L_a} - B_{_a} (\Omega_{J_a} - \Omega_{Jf_a}) - K_{o_a} (\theta_{J_a} - \theta_{Jo_a}) - B_{o_a} (\Omega_{J_a} - \Omega_{Jo_a}) \right. \\
&\quad \left. - K_{m_a} (\theta_{J_a} - \theta_{Jm_a}) - B_{m_a} (\Omega_{J_a} - \Omega_{Jm_a}) \right] \\
\frac{d\Omega_{Jf_a}}{dt} &= \frac{1}{J_{f_a}} \left[\frac{-1}{K_{a_a}} i_{L_a} + B_{_a} (\Omega_{J_a} - \Omega_{Jf_a}) - K_{f_a} \theta_{Jf_a} - B_{f_a} \Omega_{Jf_a} + T_{s3} \right] \\
\frac{d\Omega_{Jo_a}}{dt} &= \frac{1}{J_{o_a}} \left[K_{o_a} (\theta_{J_a} - \theta_{Jo_a}) + B_{o_a} (\Omega_{J_a} - \Omega_{Jo_a}) + T_{s1_a} + T_{s4_a} \right] \\
\frac{d\Omega_{Jm_a}}{dt} &= \frac{1}{J_{m_a}} \left[K_{m_a} (\theta_{J_a} - \theta_{Jm_a}) + B_{m_a} (\Omega_{J_a} - \Omega_{Jm_a}) + T_{s2_a} + T_{s5_a} \right] \\
\frac{di_{L_a}}{dt} &= \frac{1}{L_{_a}} (V_s - Ri_L - \frac{1}{Ka} (\Omega_{J_a} - \Omega_{Jf_a}))
\end{aligned}$$

2.4.3: State-space Model of Altitude System

8 states variables are the minimum set of variables which can be used to describe the altitude system. System outputs are system variables that are accessible and can be measured by the sensors (shown in Equation 26: Altitude Model Output). There are 3 system outputs we can obtain from sensors. They can generate 3 output equations. 8 state variables are substituted with 9 state variables. These 9 state equations and output equations are rewritten in state-space format (shown in Equation 24).

Equation 24: State Space Model for Altitude

$$\begin{aligned}\dot{X}_{-a} &= A_{-a9X9}X_{-a9X1} + B_{-a9X1}U_{-a1X1} + E_{-a9X5}d_{-a5X1} \\ \dot{Y}_{-a} &= C_{-a3X9}X_{-a9X1} + D_{-a3X1}U_{-a1X1}\end{aligned}$$

The system input of altitude model is voltage sent to the electric system (Shown in Equation 24).

Equation 25: Altitude Model Input

$$U = \begin{bmatrix} V_{s-a} \end{bmatrix}_{1X1}$$

There are 3 system outputs. Angular position and angular velocity of motor could be obtained from altitude encoder. Y_{t-a} is the output from star image sensor. It is the displacement of reference star.

Equation 26: Altitude Model Output

$$Y = \begin{bmatrix} \theta_{J-a} \\ \Omega_{J-a} \\ Y_{t-a} \end{bmatrix}_{3X1}$$

Y_{t-a} is a function of mirror mount angular position and OTA angular position of the azimuth system (shown in Figure 10). Its relation is listed in Equation 27. Using small angle approximation, Equation 27 can be linearized into a simpler format (shown in Equation 29). The equation is different with the azimuth model. Fork vibration affects altitude pointing.\

Equation 27: Star Image Displacement Calculation for altitude model

$$Y_{t_a} = L_{OTA} \times \sin\{2[(\theta_{Jm_a} - \theta_{J_a}) + \theta_{Jf_a}]\} - L_{OTA} \times \sin[(\theta_{Jo_a} - \theta_{J_a}) + \theta_{Jf_a}]$$

Equation 28: Reduced Star Image Displacement Equation of altitude model

$$Y_{t_a} = -L_{OTA}\theta_{J_a} + L_{OTA}\theta_{Jf_a} - L_{OTA}\theta_{Jo_a} + 2L_{OTA}\theta_{Jm_a}$$

There are five disturbance sources in the altitude model (Shown in Equation 29). The wind torque applied on OTA, the wind torque applied on mirror mount and the wind torque applied on the fork. Another two disturbances are gravitational disturbance on OTA and mirror mount.

Equation 29: Altitude Model Disturbance Source

$$d = \begin{bmatrix} T_{s1_a} \\ T_{s2_a} \\ T_{s3_a} \\ T_{s4_a} \\ T_{s5_a} \end{bmatrix}_{5 \times 1}$$

All system matrices can be determined by transformer 9 state equations and 3 output equations. System matrices are shown below. A_a matrix shown in Equation 14 is the state matrix of the altitude system.

Equation 30: System (A_a) matrix for altitude model

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{(K_{O_{-a}} + K_{m_{-a}})}{J_{-a}} & 0 & \frac{K_{O_{-a}}}{J_{-a}} & \frac{K_{m_{-a}}}{J_{-a}} & -\frac{(B_{-a} + B_{O_{-a}} + B_{m_{-a}})}{J_{-a}} & \frac{B_{-a}}{J_{-a}} & \frac{B_{O_{-a}}}{J_{-a}} & \frac{B_{m_{-a}}}{J_{-a}} & \frac{1}{K_{a_{-a}} J_{-a}} \\ 0 & -\frac{K_{f_{-a}}}{J_{f_{-a}}} & 0 & 0 & \frac{B_{-a}}{J_{f_{-a}}} & -\frac{(B_{-a} + B_{f_{-a}})}{J_{f_{-a}}} & 0 & 0 & -\frac{1}{K_{a_{-a}} J_{f_{-a}}} \\ \frac{K_{O_{-a}}}{J_{O_{-a}}} & 0 & -\frac{K_{O_{-a}}}{J_{O_{-a}}} & 0 & \frac{B_{O_{-a}}}{J_{O_{-a}}} & 0 & -\frac{B_{O_{-a}}}{J_{O_{-a}}} & 0 & 0 \\ \frac{K_{m_{-a}}}{J_{m_{-a}}} & 0 & 0 & -\frac{K_{m_{-a}}}{J_{m_{-a}}} & \frac{B_{m_{-a}}}{J_{m_{-a}}} & 0 & 0 & -\frac{B_{m_{-a}}}{J_{m_{-a}}} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{K_{a_{-a}} L_{-a}} & \frac{1}{K_{a_{-a}} L_{-a}} & 0 & 0 & -\frac{R_{-a}}{L_{-a}} \end{bmatrix}_{9 \times 9}$$

B_a matrix shown in Equation 31 is the input matrix of the altitude system.

Equation 31: Input (B_a) matrix for altitude model

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L_{-a}} \end{bmatrix}_{9 \times 1}$$

C_a matrix shown in Equation 32 is the output matrix of the altitude system. Output variables $\theta_{J_{-a}}$ and $\Omega_{J_{-a}}$ is a direction output from state variables.

$Y_{t_{-a}}$ is calculated using Equation 28.

Equation 32: Output (C_a) matrix for altitude model

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{3 \times 9}$$

D_a matrix shown in Equation 33 is the direct transmittance matrix for the altitude model. In the altitude system, input of the system doesn't affect outputs; therefore the transmittance matrix is a zero matrix.

Equation 33: Direct transmittance (D_a) matrix for altitude model

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{3 \times 1}$$

E_a matrix shown in Equation 18 is the disturbance matrix for the altitude system.

Equation 34: Disturbance (E_a) matrix for altitude model

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J_f} & 0 & 0 \\ \frac{1}{J_o} & 0 & 0 & \frac{1}{J_o} & 0 \\ 0 & \frac{1}{J_m} & 0 & 0 & \frac{1}{J_m} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{9 \times 5}$$

2.5: Matlab Simulation

2.5.1: Simulink Model

In order to simulate the telescope motion, the state-space formulas are imported into Simulink. A state space subsystem is built for handling the state space formula. The state-space subsystem is shown in Figure 13. This subsystem integrates the change rate of last step and adds the saved current state position. This forms the current state position. The current position and input are used to calculate the change rate of the current step and then import into the next step.

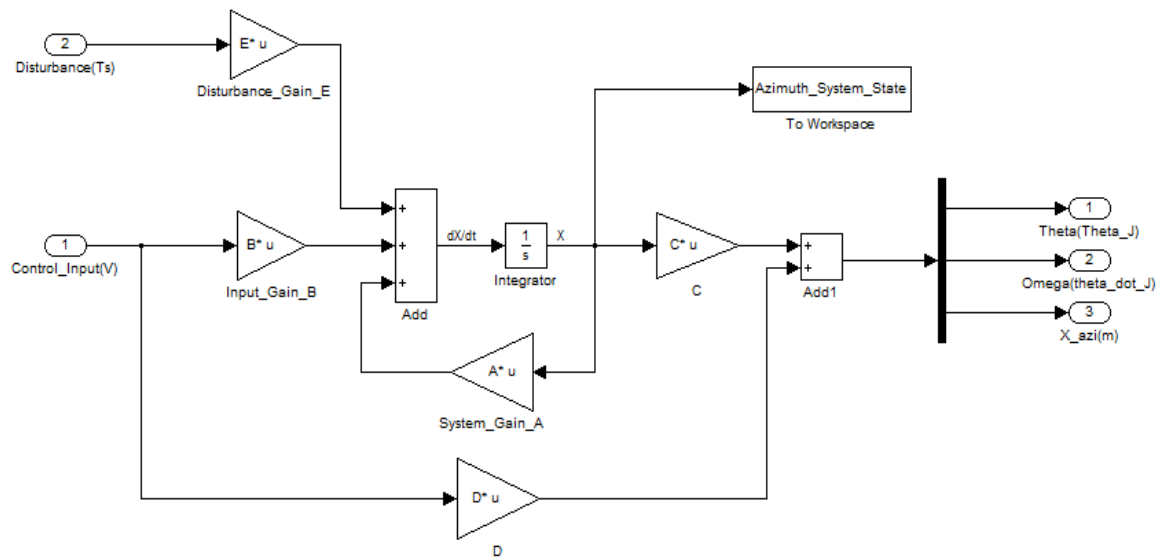


Figure 13: State-space subsystem

Gravity disturbances and wind disturbance is calculated with environmental wind speed and states of telescope. Because of the models are nonlinear, they are calculated externally in a different subsystem and imported into the state-space loop as disturbance inputs. Gravity model is shown in Figure 14. Wind model for altitude is shown in Figure 15.

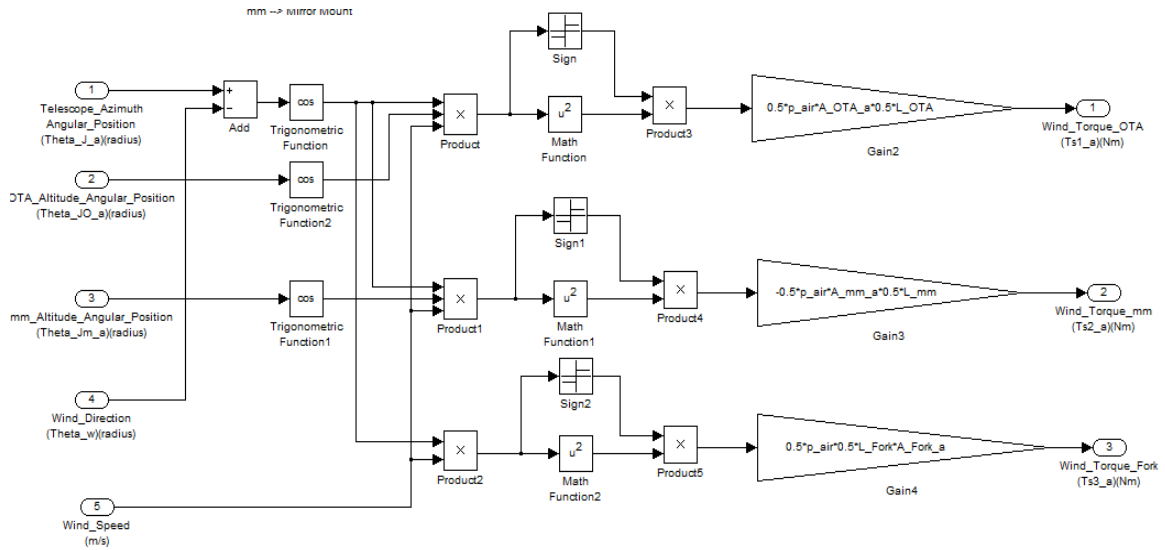


Figure 14: Wind model for altitude system

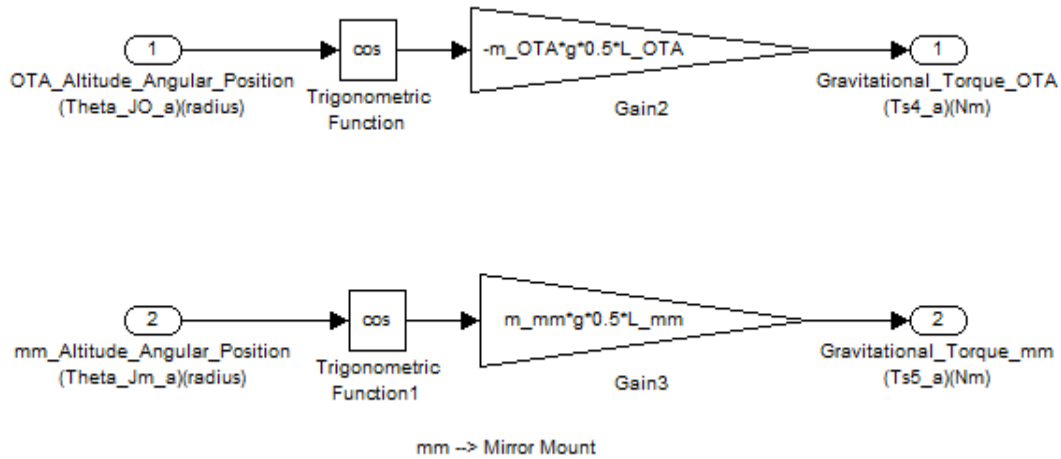


Figure 15: Gravity model

The full flexible telescope simulator is shown in Figure 16. It consists of a two state-space handler for the azimuth system and the altitude system and a disturbance model. Disturbance model includes gravity model and two wind models for the azimuth model and the altitude model. A complete set of

telescope Simulink model can be found in Appendix C.

Disturbance model takes wind direction, wind speed and telescope pose from state variables as input. It generates seven disturbance outputs and sent to the azimuth and the altitude model.

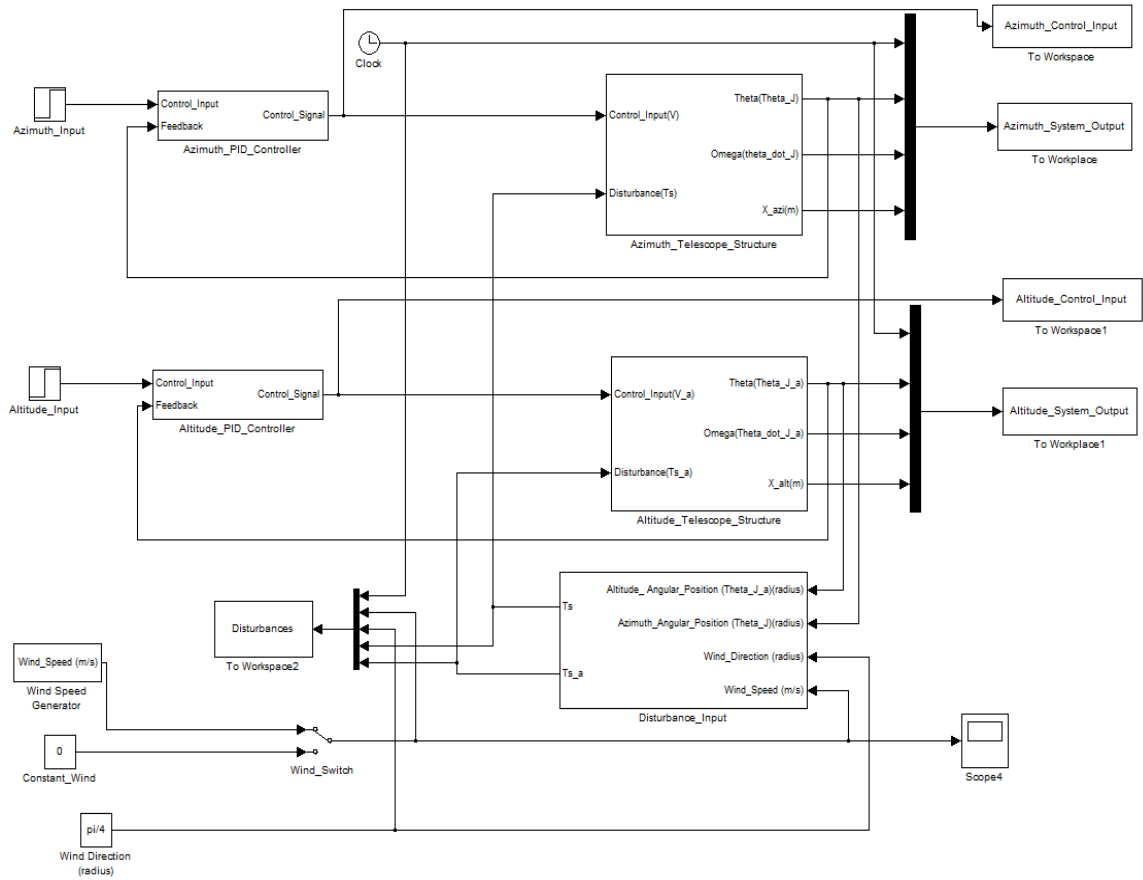


Figure 16: Full telescope in Simulink

Two PID controllers are added into the system to perform position feedback control using angular position outputs from the azimuth model and the altitude model. PID control is shown in Figure 17.

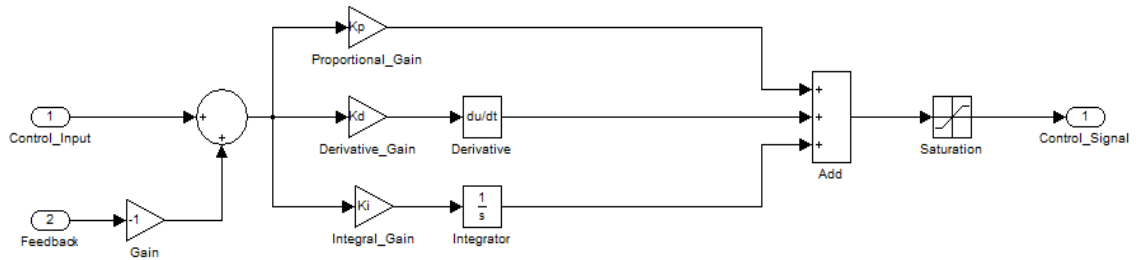


Figure 17: PID controller

2.5.2: Simulation Output

The result from the simulation is saved in Matlab. There are three Matlab plotting functions for handling a saved result. "Plot_Function_Azimuth.m" is used to plot azimuth motion. The result is shown in Figure 18. The top plot on the left shows the telescope point at the rotational axis. The mid plot on the left shows feedback from star sensor, which represents the combination telescope pointing. The bottom plot on the left shows the control input sent to telescope. Three plots on the right shows vibration of fork, OTA and mirror mount of azimuth system. The vibration is defined as difference between component angular position and main telescope pointing direction.

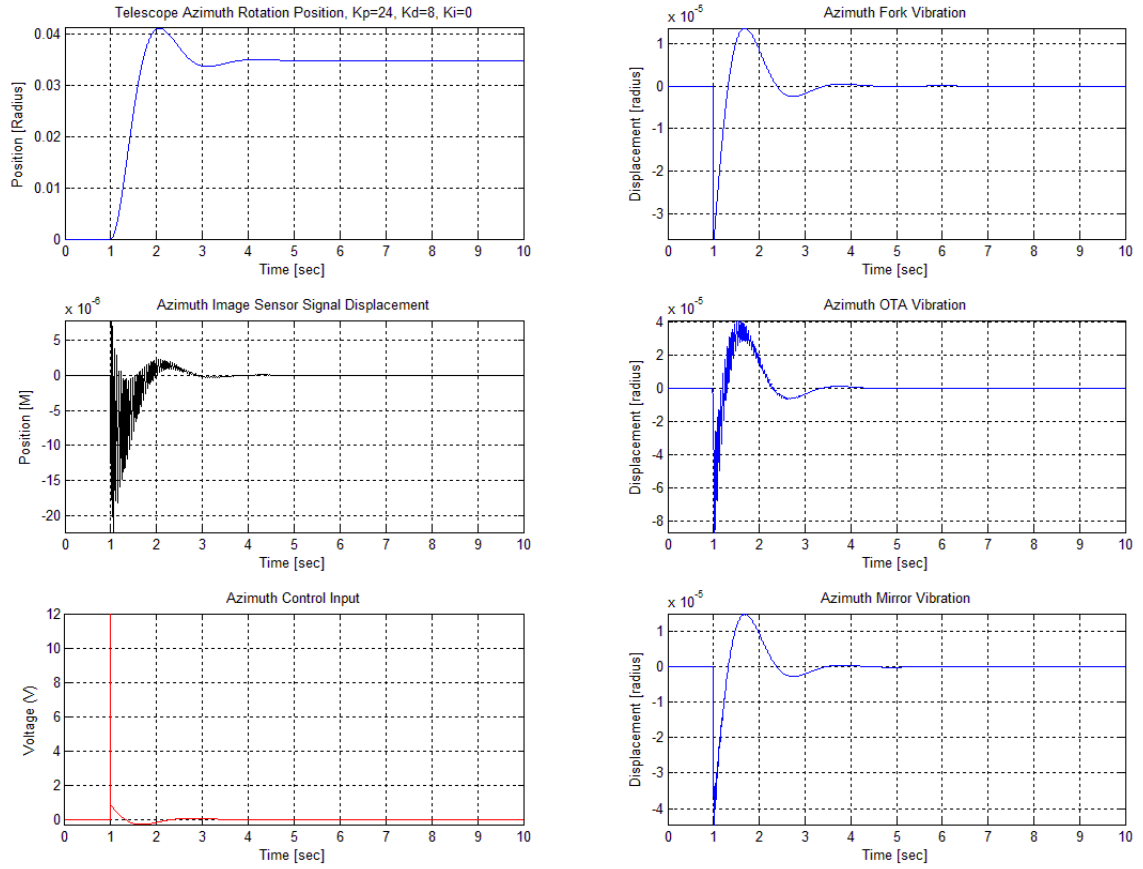


Figure 18: Result of azimuth simulation

“Plot_Function_Altitude.m” is used to plot altitude motion. The result is shown in Figure 19. It plots same information as azimuth motion.

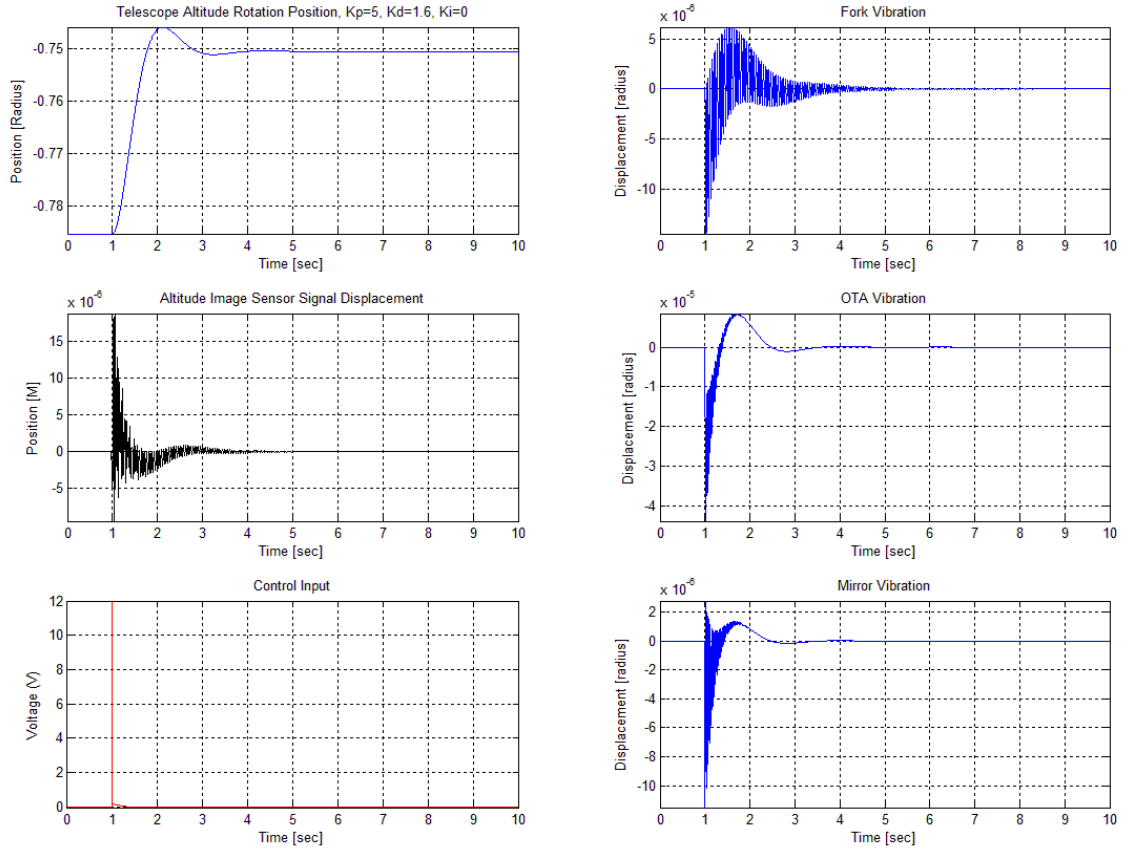


Figure 19: Result of altitude Simulation

“Plot_Function_2Axis” shows displace on two main axes and feedback on star image sensor. This information is actually observable to the control system. It also plots the change path of combined telescope pointing direction and star image feedback on XY coordination during telescope motion. The result is shown in Figure 20. The detail plotting function can be found in Appendix E.

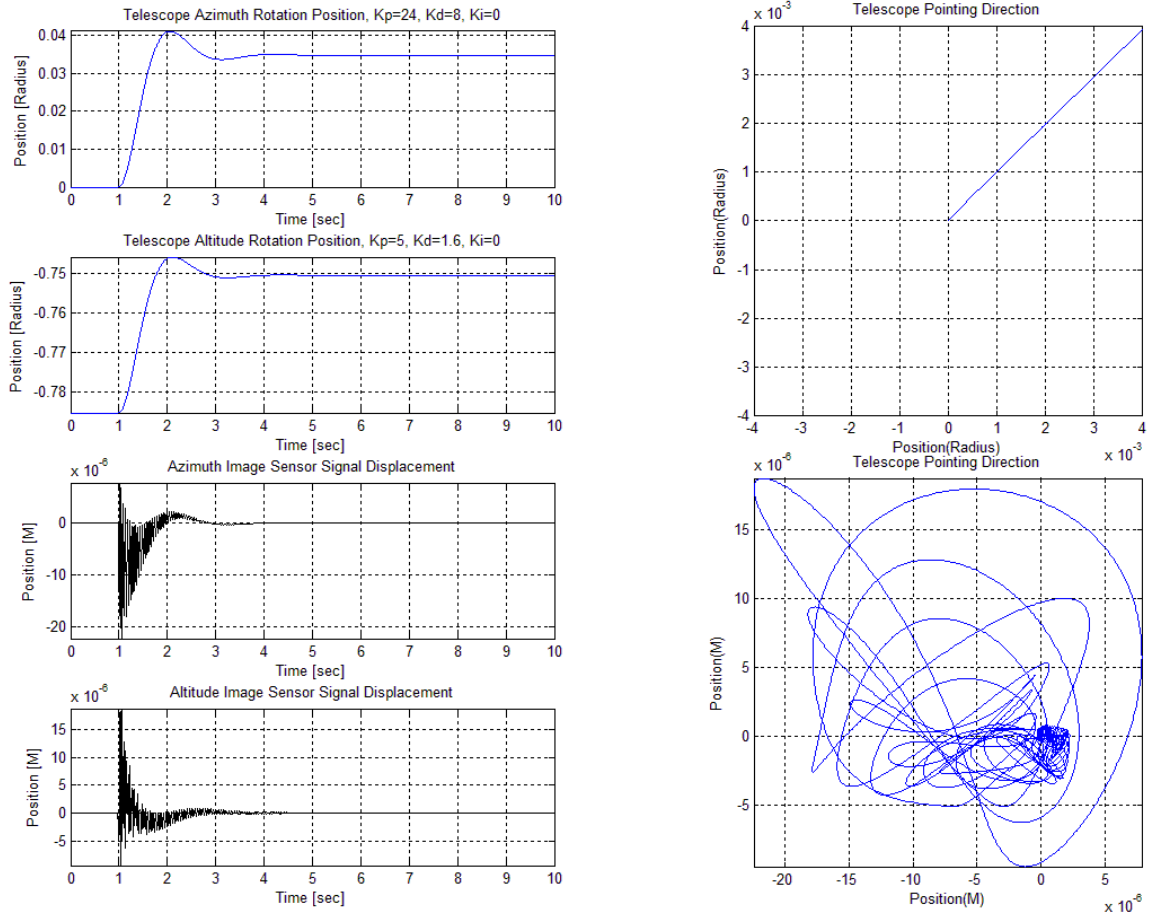


Figure 20: pointing path and star image feedback

The telescope pointing direction from the image feedback sensor shows a higher frequency motion than the pointing direction obtained from encoder. This is because the image feedback sensor also reflects the vibration in the OTA and the mirror mount.

Chapter 3: Azimuth Experimental Telescope

In order to verify the simulation, a one degree of freedom experimental telescope is built to generate a set of experimental data. The simulation parameters are adjusted so that the simulation result can reassemble this experimental data.

3.1: Experiment Set Up

3.1.1: Telescope Structure

The experimental test stand reproduces the azimuth model. The telescope experimental test stand is shown in Figure 21. The frame of experimental test stand is considered as part of the ground and not considered in the model. A Pittman 14203 motor with an E60 encoder is mounted at the bottom plate between two beams of the frame. The telescope body, composited of long steel plates, is mounted between the top and bottom plates. In order to reduce friction, two shaft adapters are manufactured using the CNC machine to mount the telescope body. The main reflector is taken down from an ORION telescope StarBlast 4.5. Its diameter is 113 mm and has a focal length of 450 mm.

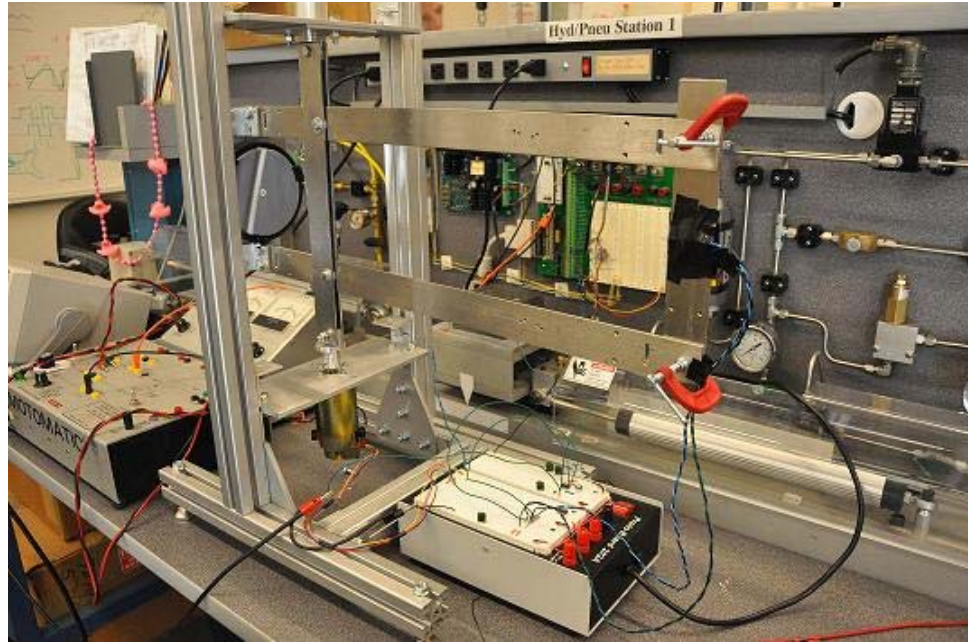


Figure 21: Experimental Setup of experimental telescope

Analog output from the controller is amplified using the MOTOMATIC Station used for Cal Poly, San Luis Obispo mechanical engineering ME 422 control class (Shown in Figure 22). The power amplifier saturated at 3 amps output. The green LED on the MOTOMATIC station turns red when power the amplifier is saturated. The power amplified has a gain of 5.25 V/V.

Derivative control is not used because the resolution of the E60 encoder attached to the Pittman motor is low. E60 encoder has 1024 pulses per revolution. Its accuracy is only 0.35 degree. Derivation of feedback error becomes infinite when encoder increment. This produces a high frequency noise in output signal when the telescope runs at high speed and produces a nonlinear system behavior during low speed. This behavior cause instability of telescope test stand; therefore, derivative gain is set to 0 for position controller of

experimental test stand.

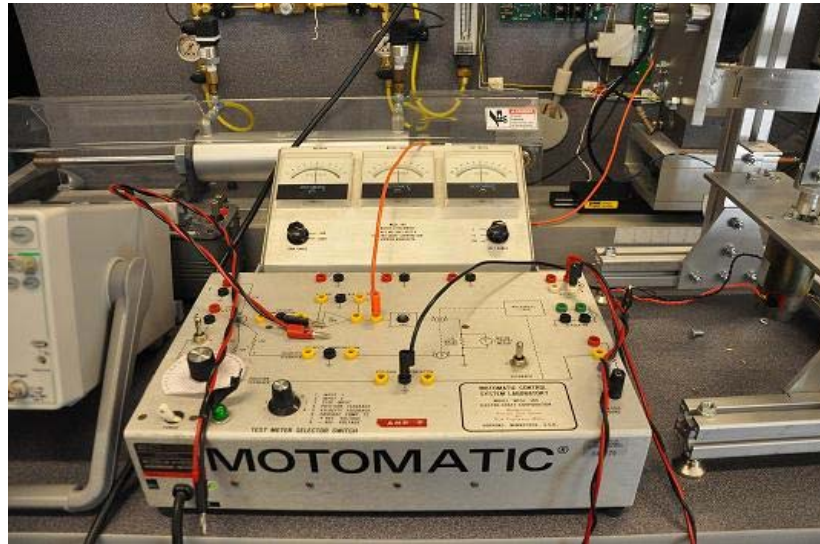


Figure 22: Motomatic Station

3.1.2: Software and Data Acquisition Interface

The experimental is controlled using a position control with PI feedback controller. The controller is constructed using Simulink. Control signals are generated by SC-2075 DAQ Interface (Data Acquisition Interface). SC-2075 DAQ interface is shown in Figure 23). This DAQ interface connects to an internal National Instrument E-Series DAQ card. The controller uses a bidirectional counter embedded in the E-Series DAQ card to determine the angular displacement of motor.

Current position controller used for the system is a single input single output controller. It takes either motor encoder reading or the star image sensors reading as input, and sends an analog signal as output. The output is connects to

power amplifier input on the MOTOMATIC station.

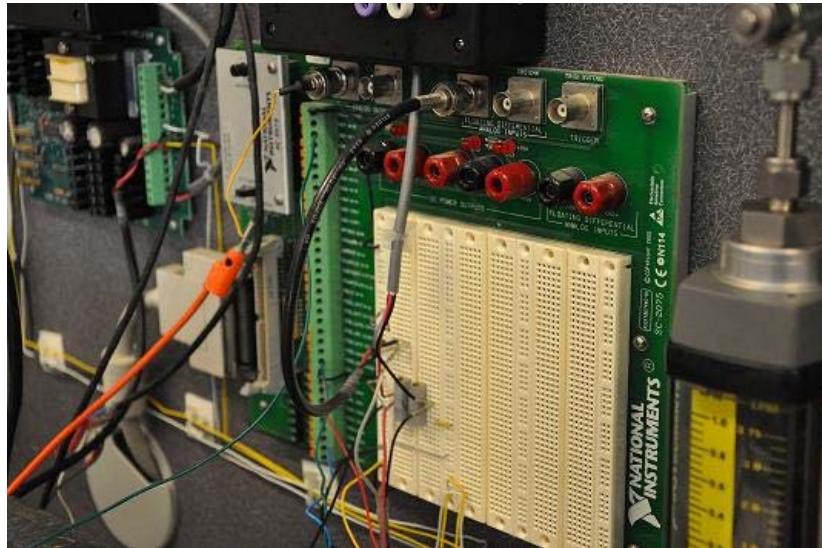


Figure 23: SC-2075 DAQ Interface

3.1.3: Guide Star Feedback Simulation

The star source is simulated with a 100W light bulb. The feedback sensor is most sensitive to infrared and red light; therefore an incandescent lamp is used for this experiment. The simulated source is shown in Figure 24.

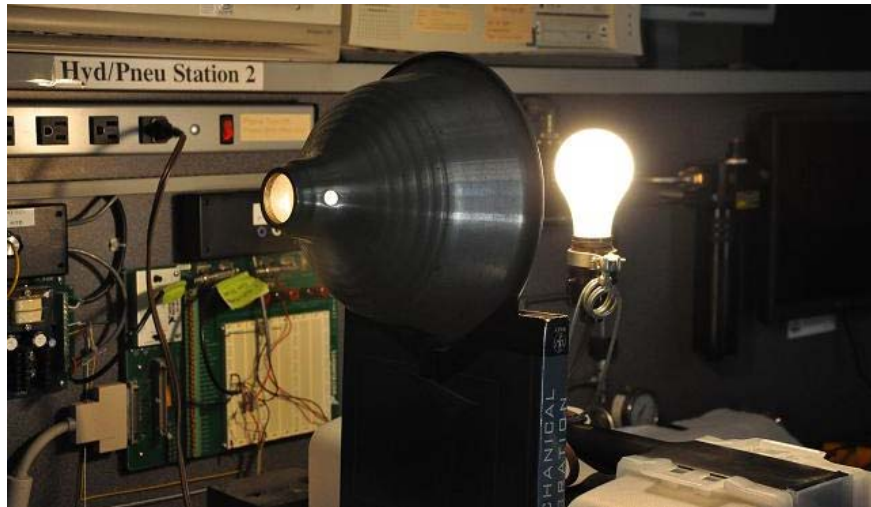


Figure 24: Simulated Star Source



Figure 25: Star Image sensor used on Test Stand

Light from the simulated source shines on the mirror and is reflected to the sensor placed close to focal point of the mirror. The star image sensor is simulated with two LED set next to each other. The distance between the two LED is adjusted so it is the same as the diameter of source light. An instrument amplifier is used to take the differential between voltage outputs of LED sensors. When the light shines on LED, the LED generates an electric current according to the intensity of the light shines on it. A light sensor used on the test stand is shown in Figure 25. The star feedback is shining on the right LED on this figure.

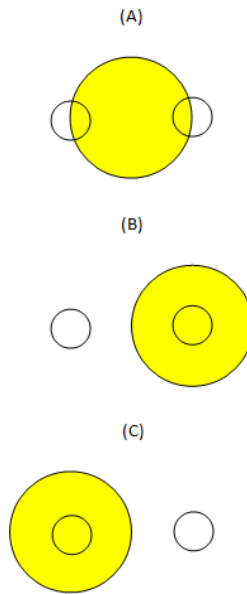


Figure 26: Star image feedback simulation sensor situation

Figure 26 shows the layout of the LED sensor. At part (A), the light source shines at the center of two sensors. Two LED send the same signal; therefore, the differential reading of two LEDs is 0. At part (B), the light source shines at the right LED. The right LED has a high reading and the left LED has low reading; therefore, the differential signal is positive. At part (C), the signal is negative.

The output from the star image sensor is shown in Figure 27. The feedback contains many high frequency noises in it. A low pass filter is applied in order to make the signal stable. In order to use this feedback in a leaner system, the center part (from 6.62 to 6.97s) of the signal is linearized. The response of the star image sensor is treated as a linear response with the relationship of the sensor output and the telescope angular position between 6.62 and 6.97 seconds. This time region is the stable region of the feedback system. If

telescope intended to use the simulated star image, the system goes unstable. For the real telescope control system, a more complex feedback system is required to judge the position of the reference star.

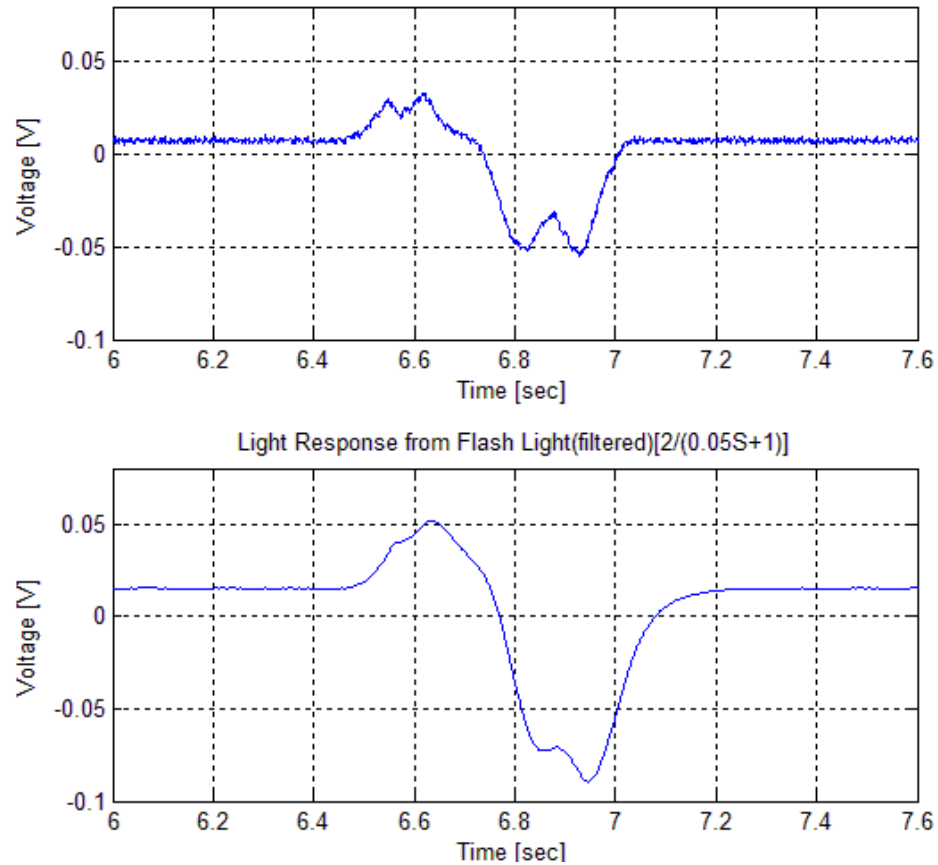


Figure 27: Light response from Simulation

3.2: Experiments Result

In order to obtain the system behavior of the experimental test stand, four tests are conducted. The testing setup is shown in Figure 28.

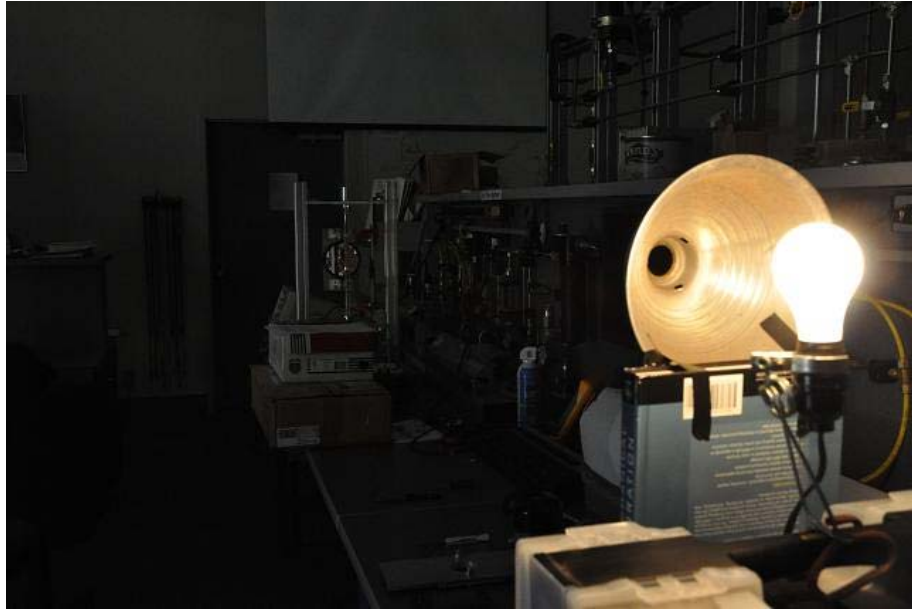


Figure 28: Telescope test

3.2.1: Light Feedback

Light feedback test is performed to find the relationship between the star image feedback and the angular position of the telescope. The experimental procedure is shown below.

1. Connect the experimental test stand to the computer and run the program “Input”. The instruction about running real time window target is shown in program interface.
2. Move telescope to find the lowest signal output.
3. Move telescope to increase signal reading and record angular displacement.
4. Repeat procedure (3) until reach highest output signal reading.

The result of the light feedback test is shown in Figure 29. This result is used to calibrate the controller gain.

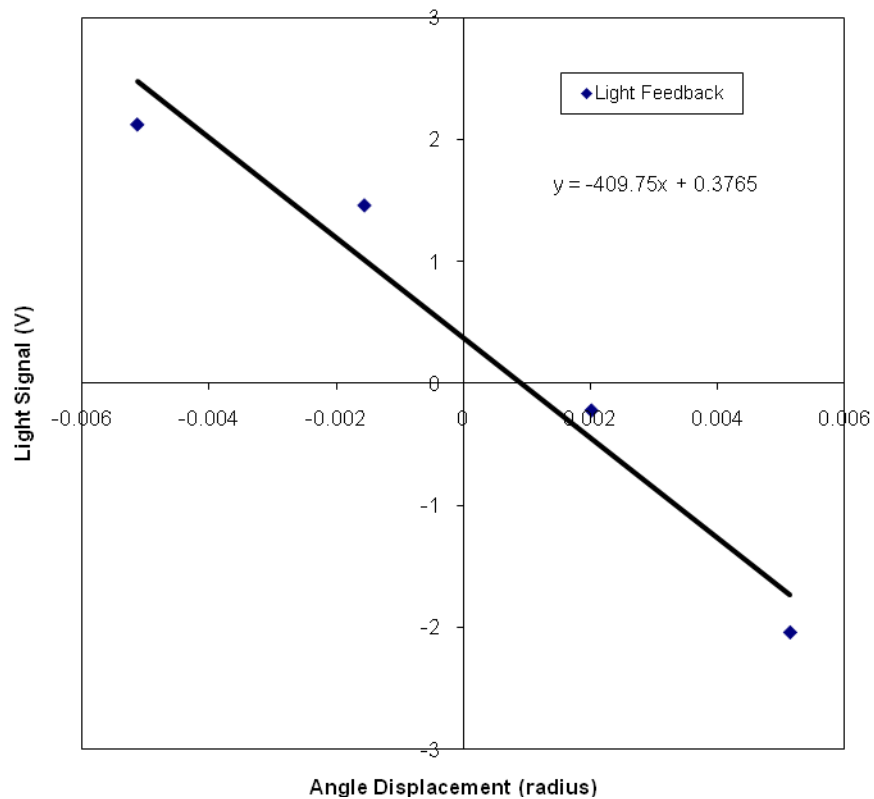


Figure 29: Star Image feedback reading

The result shows that the relation between signal output and the angular position of the telescope is close to linear; however, there is a small voltage offset. When the sensor points away from the simulated source, which the differential reading should be 0, the reading is about 20 mV.

3.2.2: PI Control

The PI control test is performed to understand telescope response using the encoder as position feedback. The experimental procedure is shown below.

1. Connect the experimental test stand to the computer and run the program "Telescope_Control". The instruction about running the real time window target is shown in program interface.
2. Adjust telescope body to center.
3. Run the program "Telescope_Control"
4. Adjust PI gain and run program again.

The PI gains that yield the best result: K_p is 7 and K_i is 0.6. The proportional gain is picked so the system has a settling time shorter than 10 sec. the integral is picked to minimize static state error. The result is shown in Figure 30. The control signal in the figure is signal sent to the power amplifier. It is required to multiply the power amplifier gain to obtain actual signal sent to motor. The response didn't actually go to the desired position. The small error toward the end is due to high static friction in the system. It is shown that the integral gain is increasing trying to compensate for that error.

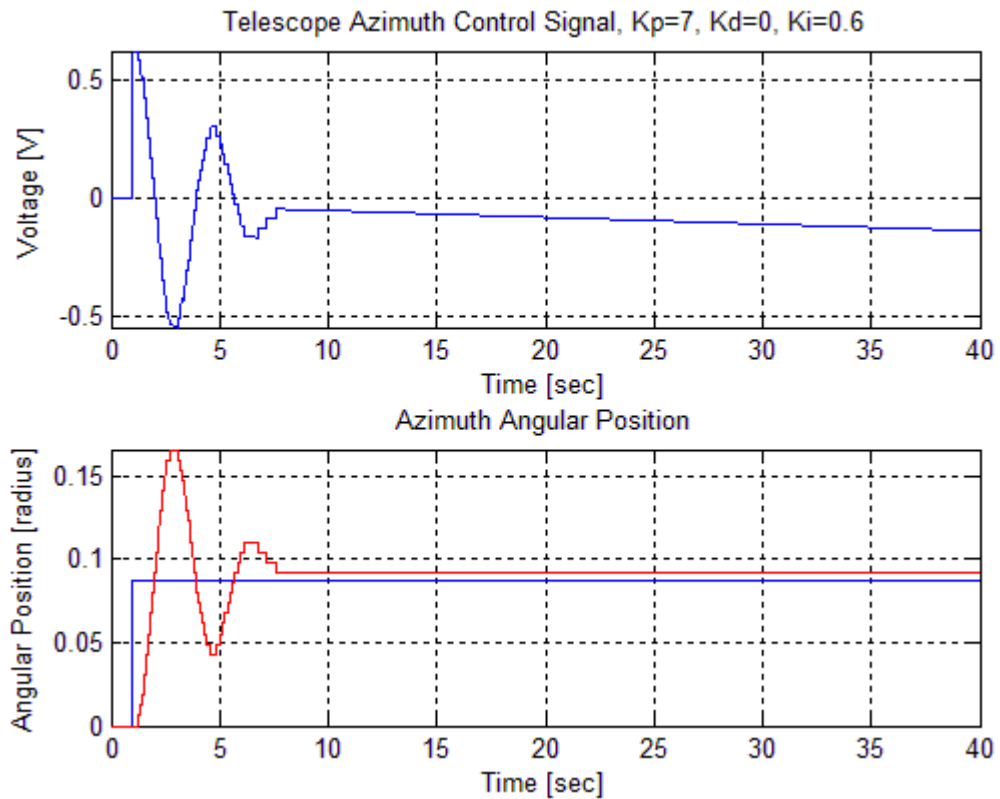


Figure 30: Telescope control response (PID control)

3.2.3: Tap Test

The tap test is performed to find the frequency of the experimental test stand. The experimental procedure is shown below.

1. Connect the experimental test stand to the computer and run the program "Input". The instruction about running real time window target is shown in the program interface.
2. Adjust the telescope body to center.
3. Clamp the center shaft of the telescope to limit its motion.

4. Gently tap the end of the OTA 3 times.
5. Record the response of the telescope sensor.

Due to the low resolution of the motor encoder, this response isn't showing in the encoder feedback. The feedback is obtained from the star image sensor. The result is shown in Figure 31. A low pass filter is applied to the signal. The result is shown in the lower part of same figure. The three peaks represent the three tap input to the system during the test. The response shows that the vibration is slowly damped out

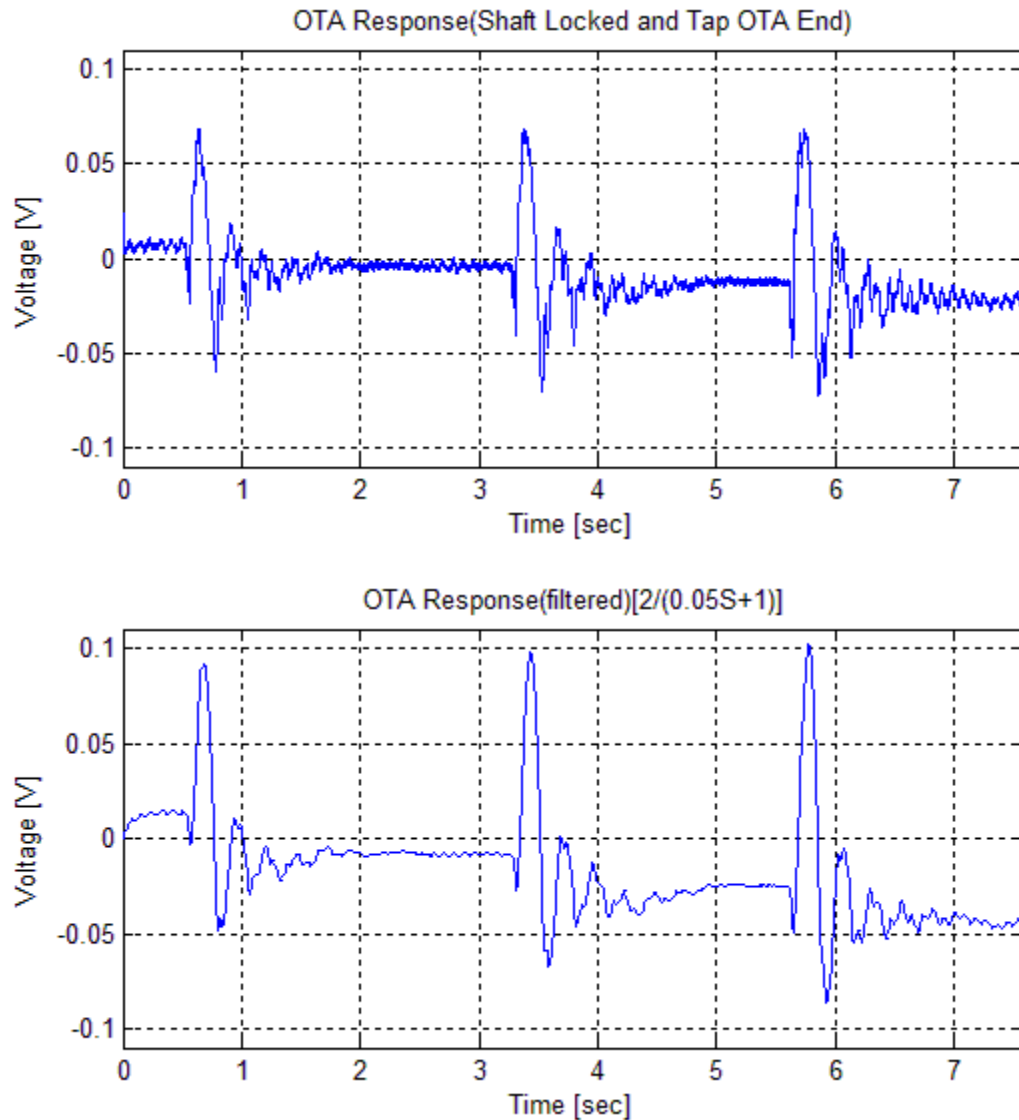


Figure 31: OTA response to tap test

In order to understand the frequency response of telescope, A FFT (fast Fourier transform) is used to analyze signal feedback. FFT function from Matlab is used for this analysis. The result is shown in Figure 32. The result shows a much higher response density on the lower frequency region. Three major response frequencies occur at 5Hz, 15Hz and 25 Hz. Two other responses show

at 60Hz and 175Hz. Further experiment is required to identify the source of each frequency.

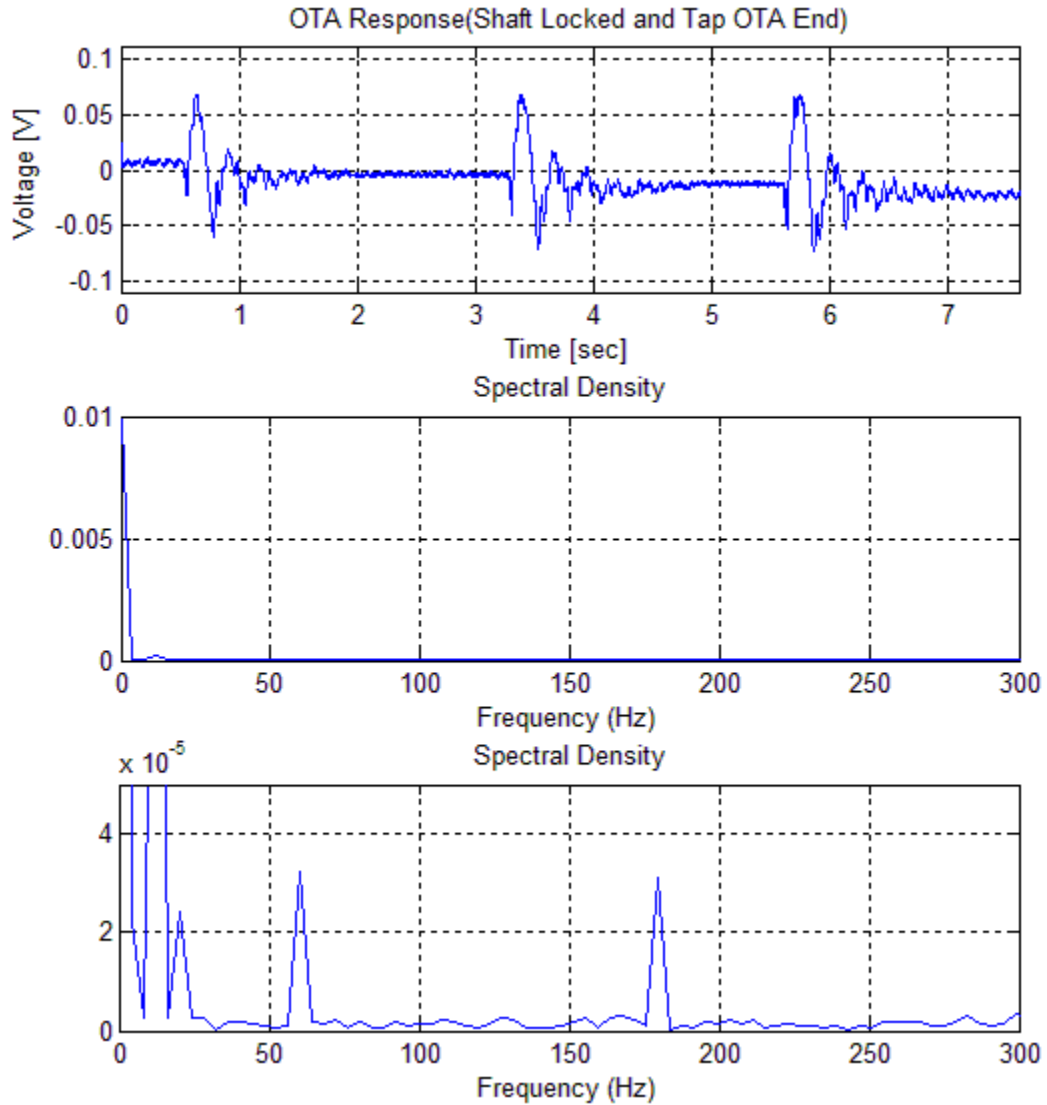


Figure 32: Frequency response of telescope to tap test

3.2.4: Feedback Control Using Light Feedback

Feedback control using light feedback is performed to understand telescope behavior when the star image sensor is used as feedback

signal. PI control is used in this test. . The experimental procedure is shown below.

1. Connect the experimental test stand to the computer and run the program “Telescope_Control_Light”. The instruction about running the real time window target is shown in the program interface.
2. Adjust the telescope body to center. Make sure source light shines at the center of two sensors.
3. Create an offset by moving telescope pointing slightly. Pointing direction shouldn't exceed linear region of feedback signals
4. Run the program “Telescope_Control_light”
5. Adjust PI gain and run program again.

The result of light feedback control is shown in Figure 33. The output signal from star image sensor is calibrated to show angular displacement in degree. Three different vibration frequencies are visible from the signal feedback. A FFT (fast Fourier transform) is used to analyze signal feedback. The result is shown in the third plot of Figure 33.

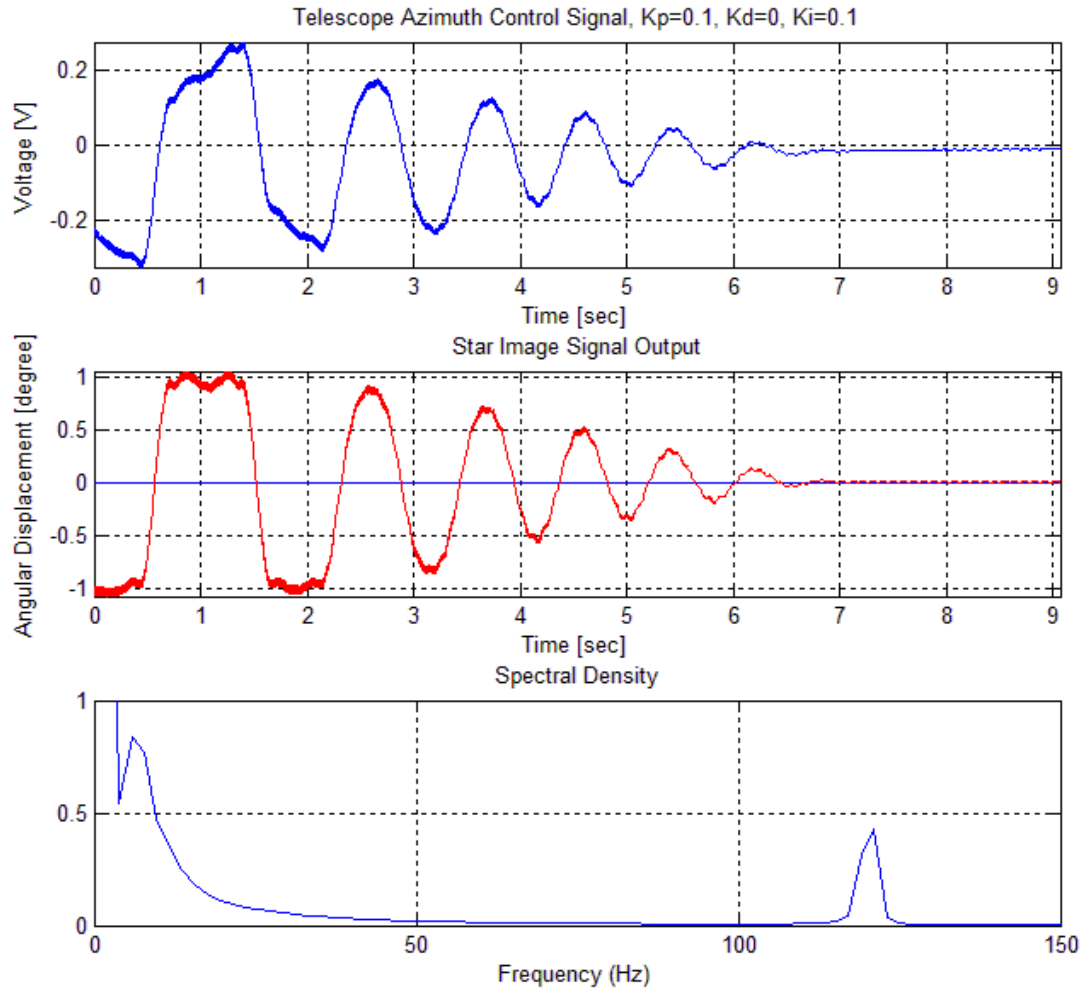


Figure 33: PI control using star image feedback

The result shows highest density toward the low frequency region. This peak comes from the oscillation of telescope motion with used control gains. The next peak shows up at 9Hz. This vibration frequency also shows up in the tap test. The source of this vibration is OTA. During the test, it is visible that OTA has a vibration motion with frequency higher than main telescope motion. Another peak shows up at 120Hz. It is not clear if it is the response from the mirror mount or the noise from AC power of the system. It is possible that the source is the

mirror mount vibration. This peak does not show in the tap test because a tap on OTA does not excite the mirror mount much. The input from control motor excites both OTA and mirror mount equally. Other frequencies where peaks occur in tap are not shown in this test because the vibration from OTA and the mirror mount is dominating. The calculation result of the natural frequency of the OTA is 7.00 Hz. The calculation result of the natural frequency of the mirror mount is 10.70 Hz. The calculation is shown in Appendix F. This result shows that the frequency is really close. Even though the mirror mount is a lot stiffer than OTA. The vibration frequency is close because the main reflector mounted on the mirror mount is a lot heavier than the instrument mounted on the OTA.

The second plot shows telescope pointing direction in degree. It shows that star image sensor works. The displacement of telescope is about 1 degree. The E60 encoder only has 3 step reading in this amount of displacement. It is hard to control small displacement using encoder as feedback source. Star image is useful for fine adjustment of telescope pointing.

Chapter 4: Simulation Result Analysis

In order to test the telescope model's ability to reassemble a real system, a similar test performed on the experimental test stand is simulated with the state space model of telescope. Full telescope model is reduced to the azimuth model, and disturbances are set to 0.

4.1: Turned Simulation model

4.1.1: PI Control Tuning

The telescope azimuth model is used to simulate PI control response of the experimental telescope. A set of general parameters is used to run the simulation first. The model is turned based on response shown in Figure 30. The PI control simulation result of turned model is shown in Figure 34. The system parameter is shown in Appendix G.

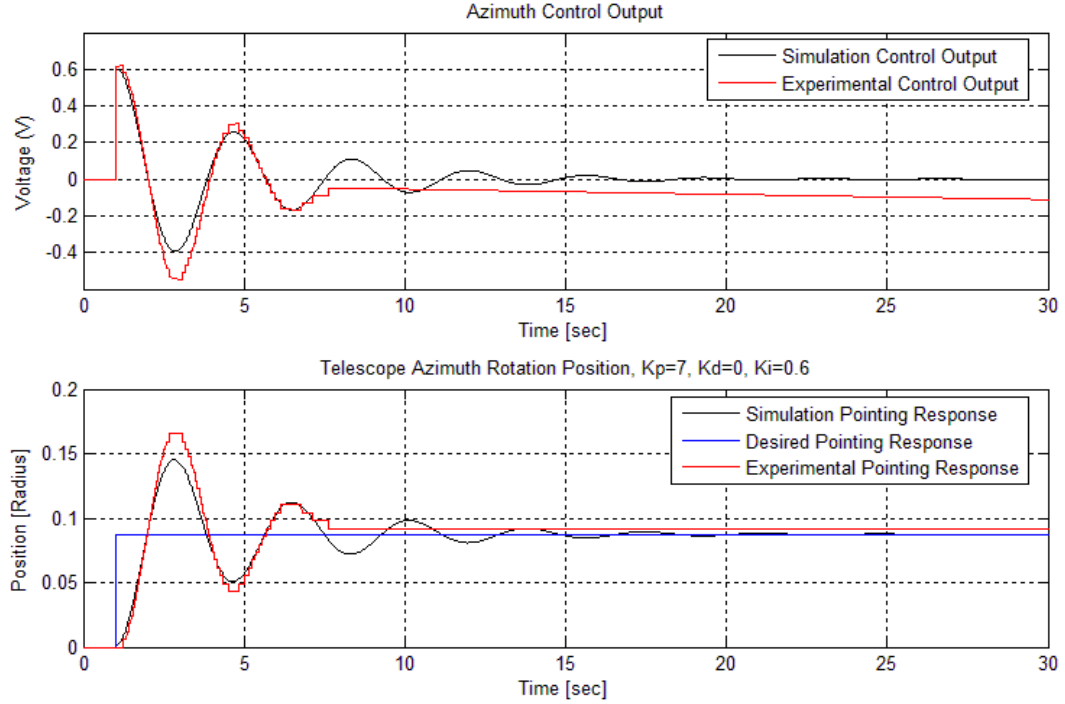


Figure 34: Turned Simulation Result

4.1.2: Frequency Response of Simulation Model

A pulse input is given to the model in order to simulate tap test. The result is shown in Figure 35. The first plot on the left is the output on the star image sensor. It shows similar patent as Figure 33. A FFT (fast Fourier transform) is used to analyze simulated signal feedback. Time domain responses of telescope components are shown on the left. 4 plots on the right show the result of FFT analysis of each component. . The result shows highest density toward the low frequency region. This peak comes from the oscillation of telescope motion with used control gains. The simulated response oscillates less; therefore, the density is smaller comparing with test result. Next peak shows up at 30Hz. The source of this is OTA vibration. This peak is also shown at OTA frequency

response plot. Next peak shows at 68Hz. The source is a mirror mount vibration. This peak is also shown at mirror mount frequency response plot. Vibrations from OTA and mirror mount project its own vibration to each other. There is a peak with small density shown on the mirror mount frequency response plot at the location where OTA vibration occurs.

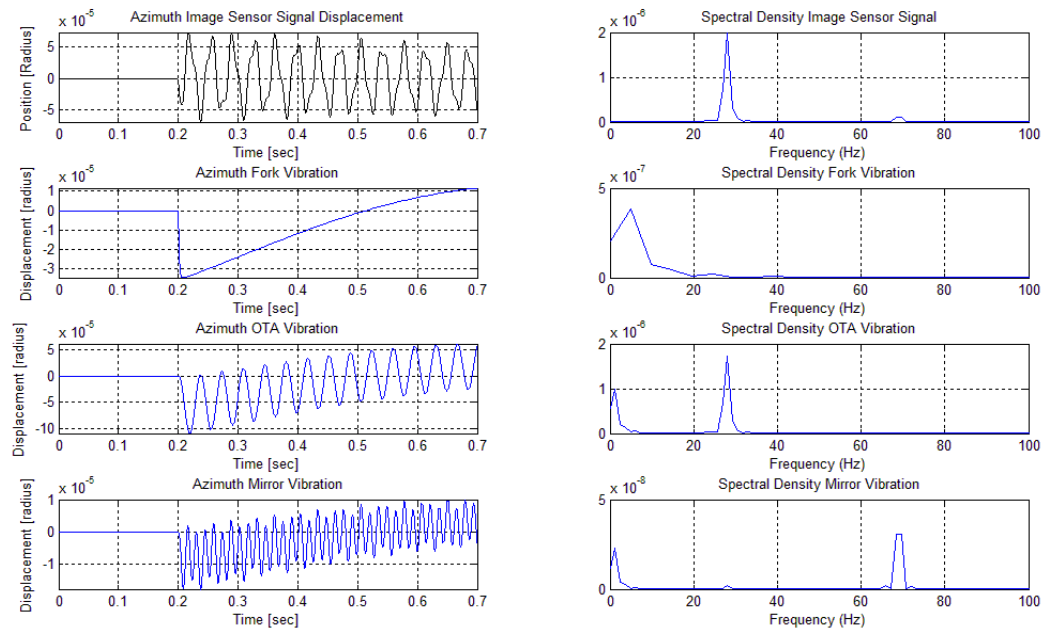


Figure 35: Frequency response of simulation model

The vibration frequency of OTA and mirror mount and OTA is show in Table 7. The natural frequencies of components match frequencies show on the frequency response plot. Natural frequency of component is adjustable by changing component mass and spring constant. The performance of simulation can be turned to fit the real system.

Table 7: Natural frequency of components

	Spring Constant	Inertia	Frequency
	[radius/Nm]	[Kgm^2]	
Mirror Mount	1806	0.0103	66.64
OTA	774	0.0298	25.65

4.2: Controllability and Observability Analysis

With system parameters that generates a simulated result that similar to real system, controllability and observability of system is analyzed. A linear system is controllable if it is possible to find a control input which can transfer a state to the origin at some finite time. A linear system is observable if its states can be determined from the output function. Controllability and observability is analyzed using criterion 2[7]. The result is shown in Table 8.

Table 8: Controllability and Observability Analysis

	Controllability	Observability
Azimuth System	3	2
Altitude System	2	2

It shows that the system is not fully controllable not fully observable. The states that are controllable and observable in the altitude system are the angular position and the angular velocity of motor: θ_{J_a} , θ_{Jf_a} . The states that are controllable in the azimuth system are the angular position and the angular velocity of the azimuth motor and the angular position of fork: θ_J , θ_{Jam} , Ω_J . The states that are observable in the azimuth system are the angular position and the angular velocity of the azimuth motor: θ_J , Ω_J . The angular position of the fork is

not observable. The star image feedback gives a combination of all vibration, but it does not show the actual position and velocity of each components.

Chapter 5: Conclusion

5.1: Light Weight Flexible Telescope Model

This project considers a light weight flexible structure telescope. By using a directive drive, it is possible to use active feedback control to reduce vibration of the telescope. This thesis fits into a larger project with a goal of controlling a flexible telescope in order to help a cheaper telescope achieve higher accuracy. In order understand the internal motion of the telescope, the goal of this thesis is to build a state space dynamic model of a two degree of freedom telescope using a known star location as feedback. The model consists of structure model with two inputs and two major environments. Two inputs are control inputs for the azimuth motor and the altitude motor. Source of disturbances are gravity and wind. The result of model is shown in Figure 16.

5.2: Experimental Test Stand

A one degree freedom experimental test stand is built to generate a series of telescope motion data in order to exam that if the telescope simulation model can represent a real telescope system. For recording the telescope motion, a wheel encoder mounted on the azimuth motor and a LED light density sensor mounted on the end of telescope body are used as feedbacks. A light feedback test is performed to find the relationship between the image sensor output and the angular position of the telescope. A tap test is performed to find the response

frequency of the experimental test stand. Two feedback position controls using the wheel encoder and the light feedback sensor are performed to observe the telescope behavior with different feedback system. The result shows that the simulation model can reproduce many system patterns shown in the experimental data; however, it is not feasible to obtain all of telescope's physical parameters from the system feedbacks.

5.3: Control Analysis

With system parameters that generates a simulated result that similar to real system, controllability and observability of system is analyzed. It shows that the system is not fully controllable or fully observable. The controllability and observability is shown in Table 8. The states that are controllable and observable in the altitude system are the angular position and the angular velocity of motor: $\theta_{J_a}, \dot{\theta}_{J_a}$. The states that are controllable in the azimuth system are the angular position and the angular velocity of the azimuth motor and the angular position of fork: $\theta_J, \dot{\theta}_{Jam}, \Omega_J$. The states that are observable in the azimuth system are the angular position and the angular velocity of the azimuth motor: $\theta_J, \dot{\theta}_{Jam}$. The angular position of the fork is not observable. The star image feedback gives a combination of all vibration, but it does not show the actual position and velocity of each components.

However, it does not mean it isn't possible to actively reduce vibration in the telescope. The current dynamic model contains three vibration

elements. In between them, the OTA causes the greatest vibration; therefore, the controller should be designed to compensate the OTA vibration. The observability is a bigger issue for controlling the system. The controller can assume the feedback from the star image sensor is caused by the OTA vibration and use it to compensate the OTA vibration. However, the experiment shows that the mirror mount could have a natural frequency close to the OTA natural frequency. It is possible that the input for reducing the OTA vibration will amplify the mirror mount vibration. To avoid this situation, more sensors are required in order to observe the system.

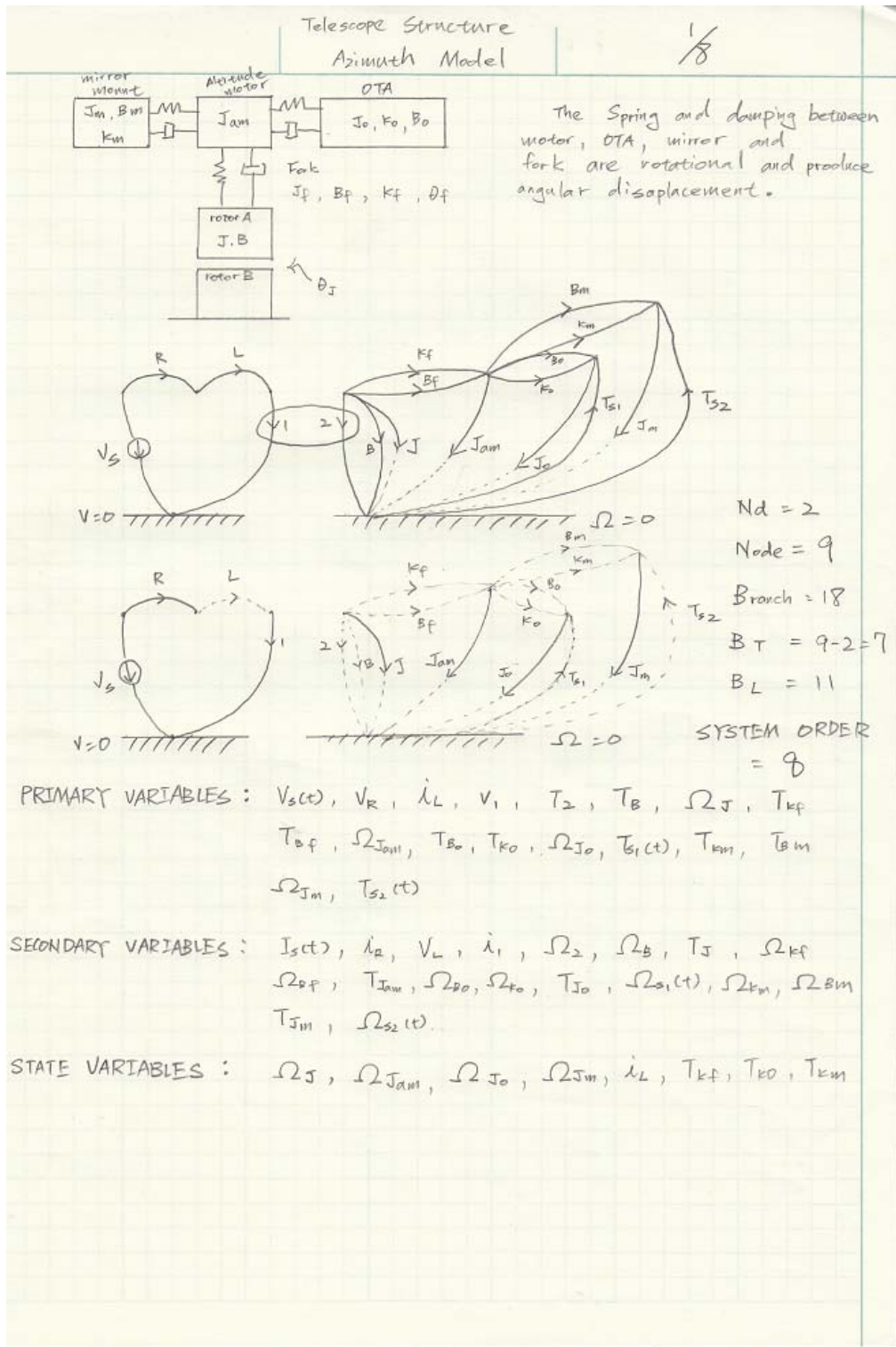
5.4: Current Project State and Future Work

In this project, a state space model of a light weight flexible telescope is built. The model will be used for the telescope simulation and the control design in the future. The state space formula also enables it to perform multiple inputs and multiple outputs control. However, due to the limitation of experimental test stand, more work should be done to understand the behavior of a light weight flexible telescope. The azimuth motor needs a wheel encoder with better resolution. The encoder should at least have 0.1 degree resolution for a better test result. A light sensor array should be built to give a bigger range of star image displacement feedback. These improvements will help understand the telescope response and build a better controller.

Reference

- [1] Karcher, H.J.; "Telescopes as mechatronic systems," *American Control Conference, 2005. Proceedings of the 2005* , vol., no., pp. 3743- 3757 vol. 6, 8-10 June 2005
- [2] Greve, A.; Karcher, H.J.; , "Performance Improvement of a Flexible Telescope Through Metrology and Active Control," *Proceedings of the IEEE* , vol.97, no.8, pp.1412-1420, Aug. 2009
- [3] Gawronski, W.; , "Modeling wind-gust disturbances for the analysis of antenna pointing accuracy," *Antennas and Propagation Magazine, IEEE* , vol.46, no.1, pp. 50- 58, Feb 2004
- [4] Molfese, C.; Schipani, P.; Capaccioli, M.; Sedmak, G.; D'Orsi, S.; , "Survey Telescope control electronics," *Power Electronics, Electrical Drives, Automation and Motion, 2008. SPEEDAM 2008. International Symposium on* , vol., no., pp.523-527, 11-13 June 2008
- [5] Derek Rowell, David N. Wormley; "System Dynamics An Introduction" Prentice Hall, Upper Saddle River, New Jersey,1997
- [6] KNMI HYDRA PROJECT < <http://www.knmi.nl/samenw/hydra/index.html>>
- [7] Willaim L. Brogan; " Modern Control Theory" Prentice Hall, Upper Saddle River, New Jersey,1991

Appendix A: Derivation of Azimuth Model



Telescope Structure
Azimuth Model

2/8

$$\frac{d\Omega_J}{dt} = \frac{1}{J} T_J$$

$$T_J = -T_z - T_B - T_{kf} - T_{Bf}$$

$$\frac{d\Omega_{Jam}}{dt} = \frac{1}{J_{Jam}} T_{Jam}$$

$$T_{Jam} = T_{kf} + T_{Bf} - T_{ko} - T_{Bo} - T_{km} - T_{Bm}$$

$$\frac{d\Omega_{Jo}}{dt} = \frac{1}{J_o} T_{Jo}$$

$$T_{Jo} = T_{ko} + T_{Bo} + T_{s1}$$

$$\frac{d\Omega_{Jm}}{dt} = \frac{1}{J_m} T_{Jm}$$

$$T_{km} = T_{km} + T_{Bm} + T_{s2}$$

$$\frac{d\dot{i}_L}{dt} = \frac{1}{L} V_L$$

$$V_L = V_s - V_R - V_1$$

$$\frac{dT_{kf}}{dt} = K_f \Omega_{kf}$$

$$\Omega_{kf} = \Omega_J - \Omega_{Jam}$$

$$\frac{dT_{ko}}{dt} = K_o \Omega_{ko}$$

$$\Omega_{ko} = \Omega_{Jam} - \Omega_{Jo}$$

$$\frac{dT_{km}}{dt} = K_m \Omega_{km}$$

$$\Omega_{km} = \Omega_{Jam} - \Omega_{Jm}$$

$$T_B = B \Omega_B = B \Omega_J$$

$$T_{Bf} = B_f \Omega_{Bf} = B_f (\Omega_J - \Omega_{Jam})$$

$$T_{Bo} = B_o \Omega_{Bo} = B_o (\Omega_{Jam} - \Omega_{Jo})$$

$$T_{Bm} = B_m \Omega_{Bm} = B_m (\Omega_{Jam} - \Omega_{Jm})$$

$$V_R = R \dot{i}_L$$

$$V_1 = \frac{1}{K_a} \Omega_2 = \frac{1}{K_a} \Omega_J$$

$$T_z = -\frac{1}{K_a} \dot{i}_L$$

Replace State with

$$\theta_J, \theta_{Jam}, \theta_{Jo}, \theta_{Jm}$$

$$\Omega_J, \Omega_{Jam}, \Omega_{Jo}, \Omega_{Jm}$$

$$\dot{i}_L$$

Replace T_{kf} , T_{ko} , T_{km}

$$T_{kf} = K_f (\theta_J - \theta_{Jam})$$

$$T_{km} = K_m (\theta_{Jam} - \theta_{Jm})$$

$$T_{ko} = K_o (\theta_{Jam} - \theta_{Jo})$$

Telescope Structure
Azimuth Model

3/8

- ① $\frac{d\theta_J}{dt} = \Omega_J$
- ② $\frac{d\theta_{Jam}}{dt} = \Omega_{Jam}$
- ③ $\frac{d\theta_{Jo}}{dt} = \Omega_{Jo}$
- ④ $\frac{d\theta_{Jm}}{dt} = \Omega_{Jm}$
- ⑤ $\frac{d\Omega_J}{dt} = \frac{1}{J} \left[\frac{1}{k_a} \dot{\lambda}_L - B\Omega_J - k_f(\theta_J - \theta_{Jam}) - B_f(\Omega_J - \Omega_{Jam}) \right]$
- ⑥ $\frac{d\Omega_{Jam}}{dt} = \frac{1}{J_m} \left[k_f(\theta_J - \theta_{Jam}) + B_f(\Omega_J - \Omega_{Jam}) - k_o(\theta_{Jam} - \theta_{Jo}) - B_o(\Omega_{Jam} - \Omega_{Jo}) - k_m(\theta_{Jm} - \theta_{Jm}) - B_m(\Omega_{Jam} - \Omega_{Jm}) \right]$
- ⑦ $\frac{d\Omega_{Jo}}{dt} = \frac{1}{J_o} \left[k_o(\theta_{Jam} - \theta_{Jo}) + B_o(\Omega_{Jam} - \Omega_{Jo}) + T_{s1}(t) \right]$
- ⑧ $\frac{d\Omega_{Jm}}{dt} = \frac{1}{J_m} \left[k_m(\theta_{Jam} - \theta_{Jm}) + B_m(\Omega_{Jam} - \Omega_{Jm}) + T_{s2}(t) \right]$
- ⑨ $\frac{d\lambda_L}{dt} = \frac{1}{L} \left[V_s - R\dot{\lambda}_L - \frac{1}{k_a} \Omega_J \right]$

Telescope Structure														
Azimuth Model														
$\frac{4}{8}$														
$\frac{d\theta_I}{dt}$	=	0	+	0	+	0	+	0	+	0	+	0	+	0
$\frac{d\theta_{Jm}}{dt}$	=	0	+	0	+	0	+	0	+	0	+	0	+	0
$\frac{d\theta_{Jo}}{dt}$	=	0	+	0	+	0	+	0	+	0	+	0	+	0
$\frac{d\theta_{Im}}{dt}$	=	0	+	0	+	0	+	0	+	0	+	0	+	0
$\frac{d\Omega_I}{dt}$	=	$-\frac{k_f}{J}\theta_I + \frac{k_f}{J}\theta_{Jm} + \frac{k_f}{J}\theta_{Jo}$	+	0	+	0	+	0	+	0	+	0	+	0
$\frac{d\Omega_{Jm}}{dt}$	=	$-\frac{k_f}{J}\theta_I + \frac{k_f}{J}\theta_{Jm} + \frac{k_f}{J}\theta_{Jo}$	+	0	+	0	+	0	+	0	+	0	+	0
$\frac{d\Omega_{Jo}}{dt}$	=	$-\frac{k_f}{J}\theta_I + \frac{k_f}{J}\theta_{Jm} + \frac{k_f}{J}\theta_{Jo}$	+	0	+	0	+	0	+	0	+	0	+	0
$\frac{d\dot{\lambda}_I}{dt}$	=	0	+	0	+	0	+	0	+	0	+	0	+	0

Telescope Structure Azimuth Model

B/8

STATE VARIABLES : $\theta_J, \theta_{Jam}, \theta_{Jo}, \theta_{Jm}, \Omega_J, \Omega_{Jam}, \Omega_{Jo}, \Omega_{Jm}, \dot{i}_L$

STATE

$$\underline{X} = \begin{bmatrix} \theta_J \\ \theta_{Jam} \\ \theta_{Jo} \\ \theta_{Jm} \\ \Omega_J \\ \Omega_{Jam} \\ \Omega_{Jo} \\ \Omega_{Jm} \\ \dot{i}_L \end{bmatrix}_{9 \times 1}$$

INPUT

$$\underline{U} = \begin{bmatrix} V_s \end{bmatrix}_{1 \times 1}$$

DISTURBANCE

$$\underline{d} = \begin{bmatrix} T_{s1} \\ T_{s2} \end{bmatrix}_{2 \times 1}$$

OUTPUT

$$\underline{Y} = \begin{bmatrix} \theta_J \\ \Omega_J \\ \chi_{azi} \end{bmatrix}_{3 \times 1}$$

$$\dot{\underline{X}}_{9 \times 1} = \underline{A}_{9 \times 9} \underline{X}_{9 \times 1} + \underline{B}_{9 \times 1} \underline{U}_{1 \times 1} + \underline{E}_{9 \times 2} \underline{d}_{2 \times 1}$$

$$\underline{Y}_{3 \times 1} = \underline{C}_{3 \times 9} \underline{X}_{9 \times 1} + \underline{D}_{3 \times 1} \underline{U}_{1 \times 1}$$

B MATRIX

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}_{9 \times 1}$$

E MATRIX

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/J_o & 0 \\ 0 & 1/J_m \\ 0 & 0 \end{bmatrix}$$

D MATRIX

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

DISTURBANCE

$$d = \begin{bmatrix} T_{s1} \\ T_{s2} \end{bmatrix}$$

T_{s1} and T_{s2} is the torque caused by wind that act on telescope body

in azimuth model, it affects Ω_{Jo} , Ω_{Jm} , effect on Ω_{Jf} is ignored, since rotation of fork due to wind is very small

Wind Model

$$\text{Wind Pressure} = \frac{1}{2} \times (\text{density of air}) \times (\text{Wind Speed})^2 \times (\text{Shape Factor})$$

V_w

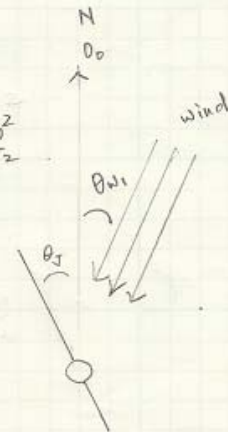
$$\text{density of air} \approx 1.25 \text{ kg/m}^3$$

$$\text{shape factor} = 1$$

$$\text{Unit of wind speed} = \frac{m}{s}$$

$$T_{s1} = \frac{1}{2} \times 1.25 \frac{\text{kg}}{\text{m}^3} \times [V_w \sin(\theta_{Jo} - \theta_{w1})]^2 \frac{\text{m}^2}{s^2} \times A_{OTA} \text{ m}^2 \times \frac{1}{2} L_{OTA} \text{ m} \quad (13)$$

$$T_{s2} = \frac{1}{2} \times 1.25 \times [V_w \sin(\theta_{Jm} - \theta_{w1})]^2 \times A_{\text{mirror mount}} \times \frac{1}{2} L_{\text{mirror mount}} \quad (14)$$



since T_{s1} and T_{s2} would make model nonlinear therefore, they would be calculated externally and input to model as T_{s1} and T_{s2}

Telescope structure
Azimuth Model

7/8

$$\begin{bmatrix}
 0 & 0 & 0 & 0 & -\frac{1}{k_L L} & 0 & 0 & 0 & -\frac{R}{L} \\
 0 & 0 & 0 & 1 & 0 & \frac{f_m}{J_m} & 0 & -\frac{B_m}{J_m} & 0 \\
 0 & 0 & 1 & 0 & 0 & \frac{B_0}{J_m} & -\frac{B_0}{J_0} & 0 & 0 \\
 0 & 1 & 0 & 0 & \frac{B_f}{J} & -\frac{(B_f+B_0+B_m)}{J_m} & \frac{B_0}{J_0} & \frac{B_m}{J_m} & 0 \\
 1 & 0 & 0 & 0 & -\frac{(B_f+B_f)}{J} & \frac{B_f}{J_m} & 0 & 0 & -\frac{1}{k_L L} \\
 0 & 0 & 0 & 0 & 0 & \frac{k_m}{J_m} & 0 & -\frac{k_m}{J_m} & 0 \\
 0 & 0 & 0 & 0 & 0 & \frac{k_0}{J_m} & -\frac{k_0}{J_0} & 0 & 0 \\
 0 & 0 & 0 & 0 & \frac{k_f}{J} & -\frac{(k_f+k_0+k_m)}{J_m} & \frac{k_0}{J_0} & \frac{k_m}{J_m} & 0 \\
 0 & 0 & 0 & 0 & -\frac{k_f}{J} & \frac{k_f}{J_m} & 0 & 0 & 0
 \end{bmatrix}$$

9x9

C MATRIX

$$\theta_J = \theta_J$$

$$\Omega_J = \Omega_J$$

x_{azi} : distance of
guide star to center
of close loop
feedback sensor

$$x_{azi} = L \sin [2(\theta_{Jm} + \theta_{Jam} - \theta_J)] + L \sin (\theta_{Jo} + \theta_{Jam} - \theta_J)$$

in order to perform controlling, x_{azi} is projected back to
global coordination to present direction telescope is actually
point

$$x_{azi} = L \sin (2(\theta_{Jm} + \theta_{Jam} - \theta_J)) - L \sin (\theta_{Jo} + \theta_{Jam} - \theta_J)$$

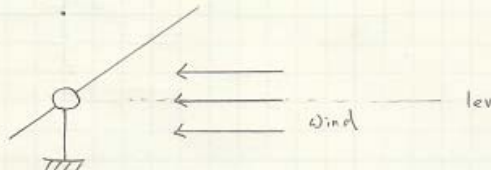
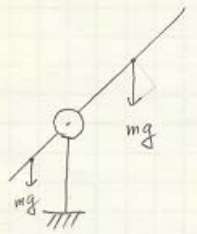
use small angle approximation.

$$x_{azi} = 3L \theta_{Jam} + L \theta_{Jo} + 2L \theta_{Jm}$$

C MATRIX

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3L & L & 2L & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Appendix B: Disturbance Model

	Telescope Structure Disturbance Model	1 / 3
Wind	 <p style="text-align: center;">Wind Pressure = $\frac{1}{2} \times (\text{density of air}) \times (\text{Wind Speed})^2 \times (\text{Shape Factor})$</p> <p style="text-align: center;">level 1</p>	V_w
	$T_{s1-a} = \frac{1}{2} \times 1.25 \times \left[V_w \cos(\theta_J - \theta_{w1}) \sin \theta_{Ja} \right]^2$ <p style="text-align: center;"> $\times A_{OTA-a} \times \frac{1}{2} L_{OTA}$ </p> <p style="text-align: center;"> \uparrow These turn is from azimuth model </p>	
	$T_{s2-a} = \frac{1}{2} \times 1.25 \times \left[V_w \cos(\theta_J - \theta_{w1}) \sin \theta_{Jm-a} \right]^2$ <p style="text-align: center;"> $\times A_{\text{mirror mount } a} \times \frac{1}{2} L_{\text{mirror mount}}$ </p>	
	$T_{s3} = \frac{1}{2} \times 1.25 \times \left[V_w \cos(\theta_J - \theta_{w1}) \right]^2$ <p style="text-align: center;"> $\times A_{\text{fork } a} \times \frac{1}{2} L_{\text{fork}}$ </p>	
Gravity	<p style="text-align: center;">Torque due to OTA</p>  <p style="text-align: center;"> $T_{s4} = -m_{OTA} \times g \times \cos \theta_{Jo-a} \times \frac{1}{2} L_{OTA}$ </p> <p style="text-align: center;"> $T_{s5} = -m_{\text{mirror mount}} \times g \times \cos \theta_{Jm-a} \times \frac{1}{2} L_{\text{mirror mount}}$ </p>	
<p>Since all the torque would make model nonlinear, they would be calculated as external disturbance source</p>		

$$d_a = \begin{bmatrix} T_{s1-a} \\ T_{s2-a} \\ T_{s3} \\ T_{s4} \\ T_{s5} \end{bmatrix}_{5 \times 1}$$

$$\frac{d\Omega_{f,a}}{dt} = \left[\text{---} + T_{s3} \right] \frac{1}{J_f}$$

$$\frac{d\Omega_{J_0,a}}{dt} = \left[\text{---} + T_{s1-a} + T_{s4} \right] \frac{1}{J_0}$$

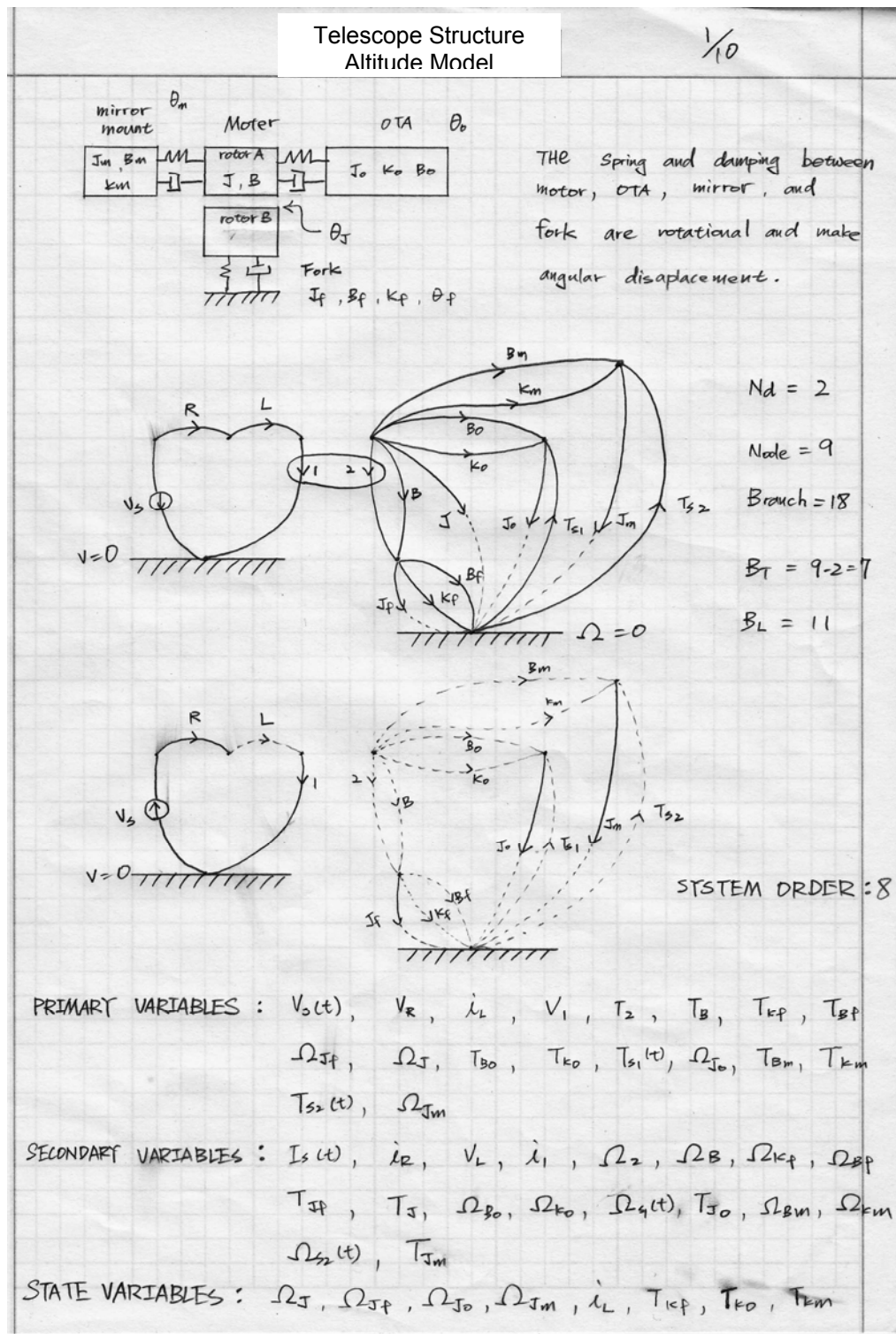
$$\frac{d\Omega_{J_m,a}}{dt} = \left[\text{---} + T_{s2-a} + T_{s5} \right] \frac{1}{J_m}$$

↑
TERM FOR STRUCTURE MATRIX

E-a matrix

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J_f} & 0 & 0 \\ \frac{1}{J_o} & 0 & 0 & \frac{1}{J_o} & 0 \\ 0 & \frac{1}{J_m} & 0 & 0 & \frac{1}{J_m} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{9 \times 5}$$

Appendix C: Derivation of Altitude Model



$$\frac{d\Omega_J}{dt} = \frac{1}{J} T_J$$

$$T_J = -T_2 - T_B - T_{k0} - T_{B0} - T_{km} - T_{Bm}$$

$$\frac{d\Omega_{Jf}}{dt} = \frac{1}{J_f} T_{Jf}$$

$$T_{Jf} = T_2 + T_B - T_{kf} - T_{Bf}$$

$$\frac{d\Omega_{J0}}{dt} = \frac{1}{J_0} T_{J0}$$

$$T_{J0} = T_{k0} + T_{B0} + T_{s1}$$

$$\frac{d\Omega_{Jm}}{dt} = \frac{1}{J_m} T_{Jm}$$

$$T_{Jm} = T_{km} + T_{Bm} + T_{s2}$$

$$\frac{d\dot{\iota}_L}{dt} = \frac{1}{L} V_L$$

$$V_L = V_s - V_R - V_i$$

$$\frac{dT_{kf}}{dt} = k_f \Omega_{kf}$$

$$\Omega_{kf} = \Omega_{Jf}$$

$$\frac{dT_{k0}}{dt} = k_0 \Omega_{k0}$$

$$\Omega_{k0} = \Omega_J - \Omega_{J0}$$

$$\frac{dT_{km}}{dt} = k_m \Omega_{km}$$

$$\Omega_{km} = \Omega_J - \Omega_{Jm}$$

$$T_B = B \Omega_B = B (\Omega_J - \Omega_{Jf})$$

$$T_{Bf} = B_f \Omega_{Bf} = B_f \Omega_{Jf}$$

$$T_{B0} = B_0 \Omega_{B0} = B_0 (\Omega_J - \Omega_{J0})$$

$$T_{Bm} = B_m \Omega_{Bm} = B_m (\Omega_J - \Omega_{Jm})$$

$$V_R = R \dot{\iota}_L$$

$$V_i = \frac{1}{k_a} \Omega_2 = \frac{1}{k_a} (\Omega_J - \Omega_{Jf})$$

$$T_2 = -\frac{1}{k_a} \dot{\iota}_L$$

3/10

$$\begin{aligned}
 \frac{d\Omega_{\text{sp}}}{dt} &= \frac{1}{J} \left[\frac{1}{K_A} \dot{\lambda}_L - B(\Omega_{\text{sp}} - \Omega_{\text{sp}0}) - T_{k0} - B_0(\Omega_{\text{sp}} - \Omega_{\text{sp}0}) - T_{km} - B_m(\Omega_{\text{sp}} - \Omega_{\text{sp}0}) \right] & \textcircled{1} \\
 \frac{d\Omega_{\text{sp}}}{dt} &= \frac{1}{J_A} \left[-\frac{1}{K_A} \dot{\lambda}_L + B(\Omega_{\text{sp}} - \Omega_{\text{sp}0}) - T_{kp} - B_p \Omega_{\text{sp}} \right] & \textcircled{2} \\
 \frac{d\Omega_{\text{sp}}}{dt} &= \frac{1}{J_0} \left[T_{k0} + B_0(\Omega_{\text{sp}} - \Omega_{\text{sp}0}) + T_{s1}(t) \right] & \textcircled{3} \\
 \frac{d\Omega_{\text{sp}}}{dt} &= \frac{1}{J_m} \left[T_{km} + B_m(\Omega_{\text{sp}} - \Omega_{\text{sp}0}) + T_{s2}(t) \right] & \textcircled{4} \\
 \frac{d\dot{\lambda}_L}{dt} &= \frac{1}{L} \left[V_s - R \dot{\lambda}_L - \frac{1}{K_A} (\Omega_{\text{sp}} - \Omega_{\text{sp}0}) \right] & \textcircled{5} \\
 \frac{dT_{kp}}{dt} &= K_p \Omega_{\text{sp}} & \textcircled{6} \\
 \frac{dT_{k0}}{dt} &= K_0 (\Omega_{\text{sp}} - \Omega_{\text{sp}0}) & \textcircled{7} \\
 \frac{dT_{km}}{dt} &= K_m (\Omega_{\text{sp}} - \Omega_{\text{sp}0}) & \textcircled{8}
 \end{aligned}$$

$$T_{kp} = k_p \theta_f$$

Replace this term into
equation ① ② ③ ④

$$T_{k0} = k_0 (\theta_J - \theta_{J0})$$

$$T_{km} = k_m (\theta_J - \theta_{Jm})$$

$$\frac{d\Omega_J}{dt} = \frac{1}{J} \left[\frac{1}{k_a} \dot{i}_L - B(\Omega_J - \Omega_{Jf}) - k_0 (\theta_J - \theta_{J0}) - B_0 (\Omega_J - \Omega_{J0}) - k_m (\theta_J - \theta_{Jm}) - B_m (\Omega_J - \Omega_{Jm}) \right] \quad \text{--- ⑨}$$

$$\frac{d\Omega_{Jf}}{dt} = \frac{1}{J_f} \left[-\frac{1}{k_a} \dot{i}_L + B(\Omega_J - \Omega_{Jf}) - k_f \theta_f - B_f \Omega_{Jf} \right] \quad \text{--- ⑩}$$

$$\frac{d\Omega_{J0}}{dt} = \frac{1}{J_0} \left[k_0 (\theta_J - \theta_{J0}) + B_0 (\Omega_J - \Omega_{J0}) + T_{s1}(t) \right] \quad \text{--- ⑪}$$

$$\frac{d\Omega_{Jm}}{dt} = \frac{1}{J_m} \left[k_m (\theta_J - \theta_{Jm}) + B_m (\Omega_J - \Omega_{Jm}) + T_{s2}(t) \right] \quad \text{--- ⑫}$$

Telescope Structure
Altitude Model

5/10

STATE VARIABLES : $\theta_J, \theta_{Jf}, \theta_{Jo}, \theta_{Jm}, \Omega_J, \Omega_{Jf}, \Omega_{Jo}, \Omega_{Jm}, \dot{\lambda}_L,$

POT EQUATION ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯

INTO STATE SPACE FORM

$$\text{STATE } \mathbf{X} = \begin{bmatrix} \theta_J \\ \theta_{Jf} \\ \theta_{Jo} \\ \theta_{Jm} \\ \Omega_J \\ \Omega_{Jf} \\ \Omega_{Jo} \\ \Omega_{Jm} \\ \dot{\lambda}_L \end{bmatrix}_{9 \times 1}$$

$$\text{INPUT } \mathbf{u} = \begin{bmatrix} V_s \end{bmatrix}_{1 \times 1}$$

$$\text{DISTURBANCE } \mathbf{d} = \begin{bmatrix} T_{s1} \\ T_{s2} \end{bmatrix}_{2 \times 1}$$

$$\text{OUTPUT } \mathbf{Y} = \begin{bmatrix} \theta_J \\ \Omega_J \\ \theta_t \end{bmatrix}_{3 \times 1}$$

$$\dot{\mathbf{X}}_{9 \times 1} = \mathbf{A}_{9 \times 9} \mathbf{X}_{9 \times 1} + \mathbf{B}_{9 \times 1} \mathbf{u}_{1 \times 1} + \mathbf{E}_{9 \times 2} \mathbf{d}_{2 \times 1}$$

$$\mathbf{Y}_{3 \times 1} = \mathbf{C}_{3 \times 9} \mathbf{X}_{9 \times 1} + \mathbf{D}_{3 \times 1} \mathbf{u}_{1 \times 1}$$

DISTURBANCE

$$d = \begin{bmatrix} T_{s1} \\ T_{s2} \end{bmatrix}$$

T_{s1} and T_{s2} is the torque caused by wind that act on telescope body

in azimuth model, it affects Ω_{Jo} , Ω_{Jm} , effect on Ω_{Jp} is ignored, since rotation of fork due to wind is very small

Wind Model

$$\text{Wind Pressure} = \frac{1}{2} \times (\text{density of air}) \times (\text{Wind Speed})^2 \times (\text{Shape Factor})$$

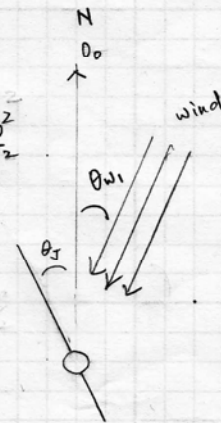
$$\text{density of air} \approx 1.25 \text{ kg/m}^3$$

$$\text{shape factor} = 1$$

$$\text{Unit of wind speed} = \frac{\text{m}}{\text{s}}$$

$$T_{s1} = \frac{1}{2} \times 1.25 \frac{\text{kg}}{\text{m}^3} \times [V_w \sin(\theta_{Jo} - \theta_{w1})]^2 \times A_{OTA} \text{ m}^2 \times \frac{1}{2} L_{OTA} \text{ m} \quad (13)$$

$$T_{s2} = \frac{1}{2} \times 1.25 \times [V_w \sin(\theta_{Jm} - \theta_{w1})]^2 \times A_{\text{mirror mount}} \times \frac{1}{2} L_{\text{mirror mount}} \quad (14)$$



since T_{s1} and T_{s2} would make model nonlinear therefore, they would be calculated externally and input to model as T_{s1} and T_{s2}

$$\frac{7}{10}$$
9 5

Telescope Structure
Altitude Model

8/10

CAMPAD

A matrix

$$\begin{bmatrix}
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 -\frac{(k_0 + k_m)}{I} & 0 & \frac{k_0}{I} & \frac{k_m}{I} & -\frac{(B_0 + B_m)}{I} & \frac{B_0}{I} & \frac{B_m}{I} & 0 & \frac{1}{k_a I} \\
 0 & \frac{I}{I_f} & 0 & 0 & \frac{B_0}{I_f} & 0 & 0 & 0 & -\frac{1}{k_a I_f} \\
 \frac{k_0}{I} & 0 & -\frac{k_0}{I} & 0 & 0 & -\frac{B_0}{I} & 0 & 0 & 0 \\
 \frac{k_m}{I} & 0 & 0 & -\frac{k_m}{I} & 0 & 0 & -\frac{B_m}{I} & 0 & 0 \\
 0 & 0 & 0 & 0 & \frac{B_0}{I} & \frac{B_m}{I} & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -\frac{1}{k_a I} & 0 & 0 & 0 & -\frac{R}{L}
 \end{bmatrix}$$

9x9

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}_{9 \times 1} \quad B \text{ MATRIX}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{3 \times 1} \quad D \text{ matrix}$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{J_0} & 0 \\ 0 & \frac{1}{J_m} \\ 0 & 0 \end{bmatrix}_{9 \times 2} \quad E \text{ matrix}$$

$$\dot{X}_{9 \times 1} = A_{9 \times 9} X_{9 \times 1} + B_{9 \times 1} u_{1 \times 1} + E_{9 \times 2} d_{2 \times 1}$$

$$Y_{3 \times 1} = C_{3 \times 9} X_{9 \times 1} + D_{3 \times 1} u_{1 \times 1}$$

C matrix

$$\theta_J = \theta_J$$

$$\Omega_J = \Omega_J$$

x_t : distance of guide
star to center of
close loop feed back
sensor

$$x_t = L \sin [2(\theta_{Jm} + \theta_{Jf})] + L \sin (\theta_{Jo} + \theta_{Jf})$$

use small angle approximation

$$x_t = 3L \theta_{Jf} + L \theta_{Jo} + 2L \theta_{Jm}$$

C matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3L & L & 2L & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

for control purpose, project x_t back to rotational
center as angle.

$$\theta_t = \theta_J + \frac{x_t}{L}$$

$$= \theta_J + 3\theta_{Jf} + \theta_{Jo} + 2L \theta_{Jm}$$

C matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 3 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Appendix D: Matlab Simulation Model

```
% this is the Matlab Simulink model "telescope_model_full"
% to use it saves it as mdl file
%
%
%
% Chi-Yeh Hsu
```

```
Model {
  Name          "telescope_model_full"
  Version       6.1
  MdlSubVersion 0
  GraphicalInterface {
    NumRootInports      0
    NumRootOutports     0
    ParameterArgumentNames ""
    ComputedModelVersion "1.254"
    NumModelReferences   0
    NumTestPointedSignals 0
  }
  SavedCharacterEncoding "ibm-1386_P100-2002"
  SaveDefaultBlockParams on
  SampleTimeColors      off
  LibraryLinkDisplay    "none"
  WideLines             off
  ShowLineDimensions    off
  ShowPortDataTypes     off
  ShowLoopsOnError      on
  IgnoreBidirectionalLines off
  ShowStorageClass      off
  ShowTestPointIcons    on
  ShowViewerIcons       on
  SortedOrder           off
  ExecutionContextIcon  off
  ShowLinearizationAnnotations on
  RecordCoverage        off
  CovPath               "/"
  CovSaveName           "covdata"
  CovMetricSettings     "dw"
  CovNameIncrementing   off
  CovHtmlReporting      on
  covSaveCumulativeToWorkspaceVar on
  CovSaveSingleToWorkspaceVar on
  CovCumulativeVarName  "covCumulativeData"
  CovCumulativeReport   off
  CovReportOnPause      on
  ScopeRefreshTime      0.035000
  OverrideScopeRefreshTime on
  DisableAllScopes      off
  DataTypeOverride       "UseLocalSettings"
  MinMaxOverflowLogging  "UseLocalSettings"
  MinMaxOverflowArchiveMode "Overwrite"
  BlockNameDataTip      off
}
```

```

BlockParametersDataTip off
BlockDescriptionStringDataTip off
ToolBar on
StatusBar on
BrowserShowLibraryLinks on
BrowserLookUnderMasks on
StopFcn "Plot_Function_Azimuth\nPlot_Function_Altitude\nPlot"
"_Function_2Axis"
Created "Thu Apr 22 12:09:46 2010"
UpdateHistory "UpdateHistoryNever"
ModifiedByFormat "%<Auto>"
LastModifiedBy "Chi-Yeh Hsu"
ModifiedDateFormat "%<Auto>"
LastModifiedDate "Thu Aug 12 14:13:28 2010"
ModelVersionFormat "1.%<AutoIncrement:254>"
ConfigurationManager "None"
LinearizationMsg "none"
Profile off
ParamWorkspaceSource "MATLABWorkspace"
AccelSystemTargetFile "accel.tlc"
AccelTemplateMakefile "accel_default_tmf"
AccelMakeCommand "make_rtw"
TryForcingSFcnDF off
ExtModeBatchMode off
ExtModeEnableFloating on
ExtModeTrigType "manual"
ExtModeTrigMode "normal"
ExtModeTrigPort "1"
ExtModeTrigElement "any"
ExtModeTrigDuration 1000
ExtModeTrigDurationFloating "auto"
ExtModeTrigHoldOff 0
ExtModeTrigDelay 0
ExtModeTrigDirection "rising"
ExtModeTrigLevel 0
ExtModeArchiveMode "off"
ExtModeAutoIncOneShot off
ExtModeIncDirWhenArm off
ExtModeAddSuffixToVar off
ExtModeWriteAllDataToWs off
ExtModeArmWhenConnect on
ExtModeSkipDownloadWhenConnect off
ExtModeLogAll on
ExtModeAutoUpdateStatusClock on
BufferReuse on
StrictBusMsg "None"
ProdHWDeviceType "32-bit Generic"
ShowModelReferenceBlockVersion off
ShowModelReferenceBlockIO off
Array {
    Type "Handle"
    Dimension 1
    Simulink.ConfigSet {
        $ObjectID 1
        Version "1.0.4"
        Array {
            Type "Handle"

```

```

Dimension          7
Simulink.SolverCC {
  $ObjectID        2
  Version           "1.0.4"
  StartTime         "0.0"
  StopTime          "t_end"
  AbsTol            "auto"
  FixedStep         "0.0001"
  InitialStep       "auto"
  MaxNumMinSteps    "-1"
  MaxOrder          5
  ExtrapolationOrder 4
  NumberNewtonIterations 1
  MaxStep           "auto"
  MinStep           "auto"
  RelTol            "1e-14"
  SolverMode        "Auto"
  Solver            "ode3"
  SolverName        "ode3"
  ZeroCrossControl  "UseLocalSettings"
  PositivePriorityOrder off
  AutoInsertRateTranBlk off
  SampleTimeConstraint "Unconstrained"
  RateTranMode      "Deterministic"
}
Simulink.DataIOCC {
  $ObjectID        3
  Version           "1.0.4"
  Decimation        "1"
  ExternalInput      "[t, u]"
  FinalStateName     "xFinal"
  InitialState       "xInitial"
  LimitDataPoints    on
  MaxDataPoints      "1000"
  LoadExternalInput  off
  LoadInitialState   off
  SaveFinalState     off
  SaveFormat         "Array"
  SaveOutput         on
  SaveState          off
  SignalLogging      on
  SaveTime           on
  StateSaveName      "xout"
  TimeSaveName       "tout"
  OutputSaveName     "yout"
  SignalLoggingName  "logsout"
  OutputOption       "RefineOutputTimes"
  OutputTimes        "[]"
  Refine             "1"
}
Simulink.OptimizationCC {
  $ObjectID        4
  Array {
    Type            "Cell"
    Dimension        5
    Cell             "ZeroExternalMemoryAtStartup"
    Cell             "ZeroInternalMemoryAtStartup"
  }
}

```

```

        Cell          "InitFltsAndDblsToZero"
        Cell          "OptimizeModelRefInitCode"
        Cell          "NoFixptDivByZeroProtection"
        PropName      "DisabledProps"
    }
    Version           "1.0.4"
    BlockReduction     on
    BooleanDataType    on
    ConditionallyExecuteInputs on
    ConditionalExecOptimization "on_for_testing"
    InlineParams       off
    InlineInvariantSignals on
    OptimizeBlockIOStorage on
    BufferReuse         on
    EnforceIntegerDowncast on
    ExpressionFolding  on
    FoldNonRolledExpr  on
    LocalBlockOutputs  on
    ParameterPooling   on
    RollThreshold      5
    SystemCodeInlineAuto off
    StateBitsets       off
    DataBitsets        off
    UseTempVars        off
    ZeroExternalMemoryAtStartup on
    ZeroInternalMemoryAtStartup on
    InitFltsAndDblsToZero on
    NoFixptDivByZeroProtection off
    OptimizeModelRefInitCode off
    LifeSpan           "inf"
}
Simulink.DebuggingCC {
    $ObjectID         5
    Version            "1.0.4"
    RTPrefix           "error"
    ConsistencyChecking "none"
    ArrayBoundsChecking "none"
    AlgebraicLoopMsg    "warning"
    ArtificialAlgebraicLoopMsg "warning"
    CheckSSInitialOutputMsg on
    CheckExecutionContextPreStartOutputMsg off
    CheckExecutionContextRuntimeOutputMsg off
    SignalResolutionControl "TryResolveAllWithWarning"
    BlockPriorityViolationMsg "warning"
    MinStepSizeMsg      "warning"
    SolverPrmCheckMsg    "warning"
    InheritedTsInSrcMsg  "warning"
    DiscreteInheritContinuousMsg "warning"
    MultiTaskDSMMsg      "warning"
    MultiTaskRateTransMsg "error"
    SingleTaskRateTransMsg "none"
    TasksWithSamePriorityMsg "warning"
    CheckMatrixSingularityMsg "none"
    IntegerOverflowMsg    "warning"
    Int32ToFloatConvMsg   "warning"
    ParameterDowncastMsg  "error"
    ParameterOverflowMsg   "error"
}

```

```

ParameterPrecisionLossMsg "warning"
UnderSpecifiedDataTypeMsg "none"
UnnecessaryDatatypeConvMsg "none"
VectorMatrixConversionMsg "none"
InvalidFcnCallConnMsg "error"
FcnCallInpInsideContextMsg "Use local settings"
SignalLabelMismatchMsg "none"
UnconnectedInputMsg "warning"
UnconnectedOutputMsg "warning"
UnconnectedLineMsg "warning"
SFcnCompatibilityMsg "none"
UniqueDataStoreMsg "none"
RootOutportRequireBusObject "warning"
AssertControl "UseLocalSettings"
EnableOverflowDetection off
ModelReferenceIOMsg "none"
ModelReferenceVersionMismatchMessage "none"
ModelReferenceIOMismatchMessage "none"
ModelReferenceCSMismatchMessage "none"
ModelReferenceSimTargetVerbose off
UnknownTsInhSupMsg "warning"
ModelReferenceDataLoggingMessage "warning"
ModelReferenceSymbolNameMessage "warning"
}
Simulink.HardwareCC {
    $ObjectID 6
    Version "1.0.4"
    ProdBitPerChar 8
    ProdBitPerShort 16
    ProdBitPerInt 32
    ProdBitPerLong 32
    ProdIntDivRoundTo "Undefined"
    ProdEndianess "Unspecified"
    ProdWordSize 32
    ProdShiftRightIntArith on
    ProdHWDeviceType "32-bit Generic"
    TargetBitPerChar 8
    TargetBitPerShort 16
    TargetBitPerInt 32
    TargetBitPerLong 32
    TargetShiftRightIntArith on
    TargetIntDivRoundTo "Undefined"
    TargetEndianess "Unspecified"
    TargetWordSize 32
    TargetTypeEmulationWarnSuppressLevel 0
    TargetPreprocMaxBitsSint 32
    TargetPreprocMaxBitsUint 32
    TargetHWDeviceType "Specified"
    TargetUnknown off
    ProdEqTarget on
}
Simulink.ModelReferenceCC {
    $ObjectID 7
    Version "1.0.4"
    UpdateModelReferenceTargets "IfOutOfDateOrStructuralChange"
    CheckModelReferenceTargetMessage "error"
}

```



```

ModelReferenceNumInstancesAllowed "Multi"
ModelReferencePassRootInputsByReference on
ModelReferenceMinAlgLoopOccurrences off
}
Simulink.RTWCC {
    $BackupClass      "Simulink.RTWCC"
    $ObjectID          8
    Array {
        Type           "Cell"
        Dimension       1
        Cell            "IncludeHyperlinkInReport"
        PropName        "DisabledProps"
    }
    Version            "1.0.4"
    SystemTargetFile    "grt.tlc"
    GenCodeOnly         off
    MakeCommand         "make_rtw"
    TemplateMakefile    "grt_default_tmf"
    GenerateReport      off
    SaveLog             off
    RTWVerbose          on
    RetainRTWFile       off
    ProfileTLC          off
    TLCDebug            off
    TLCCoverage         off
    TLCAssert           off
    ProcessScriptMode   "Default"
    ConfigurationMode   "Optimized"
    ConfigAtBuild       off
    IncludeHyperlinkInReport off
    LaunchReport        off
    Array {
        Type           "Handle"
        Dimension       2
        Simulink.CodeAppCC {
            $ObjectID    9
            Array {
                Type      "Cell"
                Dimension  9
                Cell       "IgnoreCustomStorageClasses"
                Cell       "InsertBlockDesc"
                Cell       "SFDataObjDesc"
                Cell       "SimulinkDataObjDesc"
                Cell       "DefineNamingRule"
                Cell       "SignalNamingRule"
                Cell       "ParamNamingRule"
                Cell       "InlinedPrmAccess"
                Cell       "CustomSymbolStr"
                PropName    "DisabledProps"
            }
            Version        "1.0.4"
            ForceParamTrailComments off
            GenerateComments on
            IgnoreCustomStorageClasses on
            IncHierarchyInIds off
            MaxIdLength     31
            PreserveName     off
        }
    }
}

```

```

PreserveNameWithParent    off
ShowEliminatedStatement  off
IncAutoGenComments        off
SimulinkDataObjDesc       off
SFDataObjDesc             off
IncDataTypeInIds          off
PrefixModelToSubsysFcnNames on
CustomSymbolStr           "$R$N$M"
MangleLength              1
DefineNamingRule           "None"
ParamNamingRule            "None"
SignalNamingRule           "None"
InsertBlockDesc           off
SimulinkBlockComments     on
EnableCustomComments      off
InlinedPrmAccess          "Literals"
ReqsInCode                off
}
Simulink.GRTTargetCC {
    $BackupClass           "Simulink.TargetCC"
    $ObjectID              10
    Array {
        Type               "Cell"
        Dimension          12
        Cell               "IncludeMdlTerminateFcn"
        Cell               "CombineOutputUpdateFcns"
        Cell               "SuppressErrorStatus"
        Cell               "ERTCustomFileBanners"
        Cell               "GenerateSampleERTMain"
        Cell               "MultiInstanceERTCode"
        Cell               "PurelyIntegerCode"
        Cell               "SupportNonFinite"
        Cell               "SupportComplex"
        Cell               "SupportAbsoluteTime"
        Cell               "SupportContinuousTime"
        Cell               "SupportNonInlinedSFcns"
        PropName           "DisabledProps"
    }
    Version                "1.0.4"
    TargetFcnLib            "ansi_tf1_tmw.mat"
    GenFloatMathFcnCalls   "ANSI_C"
    UtilityFuncGeneration  "Auto"
    GenerateFullHeader     on
    GenerateSampleERTMain  off
    IsPILTarget            off
    ModelReferenceCompliant on
    IncludeMdlTerminateFcn on
    CombineOutputUpdateFcns off
    SuppressErrorStatus    off
    IncludeFileDelimiter   "Auto"
    ERTCustomFileBanners  off
    SupportAbsoluteTime    on
    LogVarNameModifier     "rt_"
    MatFileLogging         on
    MultiInstanceERTCode   off
    SupportNonFinite       on
    SupportComplex          on

```

```

        PurelyIntegerCode      off
        SupportContinuousTime  on
        SupportNonInlinedSFCns on
        ExtMode                 off
        ExtModeStaticAlloc      off
        ExtModeTesting           off
        ExtModeStaticAllocSize  1000000
        ExtModeTransport        0
        ExtModeMexFile          "ext_comm"
        RTWCAPISignals          off
        RTWCAPIParams           off
        RTWCAPISStates          off
        GenerateASAP2           off
    }
    PropName                    "Components"
}
}
PropName                      "Components"
}
    Name                       "Configuration"
    SimulationMode              "normal"
    CurrentDlgPage              "Solver"
}
PropName                      "ConfigurationSets"
}
Simulink.ConfigSet {
    $PropName                   "ActiveConfigurationSet"
    $ObjectID                   1
}
BlockDefaults {
    Orientation                  "right"
    ForegroundColor              "black"
    BackgroundColor              "white"
    DropShadow                   off
    NamePlacement                "normal"
    FontName                     "Helvetica"
    FontSize                     10
    FontWeight                   "normal"
    FontAngle                    "normal"
    ShowName                     on
}
BlockParameterDefaults {
    Block {
        BlockType                Clock
        DisplayTime               off
    }
    Block {
        BlockType                Constant
        Value                     "1"
        VectorParams1D            on
        OutDataTypeMode           "Inherit from 'Constant value'"
        OutDataType               "sfix(16)"
        ConRadixGroup             "Use specified scaling"
        OutScaling                "2^0"
        SampleTime                "inf"
    }
    Block {

```

```

        BlockType      Demux
        Outputs         "4"
        DisplayOption   "none"
        BusSelectionMode off
    }
    Block {
        BlockType      Derivative
        LinearizePole   "inf"
    }
    Block {
        BlockType      DiscretePulseGenerator
        PulseType       "Sample based"
        TimeSource      "Use simulation time"
        Amplitude       "1"
        Period          "2"
        PulseWidth      "1"
        PhaseDelay      "0"
        SampleTime      "1"
        VectorParams1D   on
    }
    Block {
        BlockType      Gain
        Gain            "1"
        Multiplication   "Element-wise(K.*u)"
        ParameterDataTypeMode "Same as input"
        ParameterDataType "sfix(16)"
        ParameterScalingMode "Best Precision: Matrix-wise"
        ParameterScaling  "2^0"
        OutDataTypeMode   "Same as input"
        OutDataType       "sfix(16)"
        OutScaling        "2^0"
        LockScale         off
        RndMeth           "Floor"
        SaturateOnIntegerOverflow on
        SampleTime        "-1"
    }
    Block {
        BlockType      SignalConversion
        OverrideOpt     off
    }
    Block {
        BlockType      Inport
        BusObject       "BusObject"
        BusOutputAsStruct off
        PortDimensions  "-1"
        SampleTime      "-1"
        DataType        "auto"
        OutDataType      "sfix(16)"
        OutScaling       "2^0"
        SignalType       "auto"
        SamplingMode     "auto"
        Interpolate      on
    }
    Block {
        BlockType      Integrator
        ExternalReset   "none"
        InitialConditionSource "internal"
    }

```

```

InitialCondition      "0"
LimitOutput           off
UpperSaturationLimit  "inf"
LowerSaturationLimit  "-inf"
ShowSaturationPort    off
ShowStatePort         off
AbsoluteTolerance     "auto"
IgnoreLimit           off
ZeroCross             on
}
Block {
  BlockType           Math
  Operator             "exp"
  OutputSignalType     "auto"
  SampleTime          "-1"
  OutDataTypeMode      "Same as first input"
  OutDataType          "sfix(16)"
  OutScaling           "2^0"
  LockScale            off
  RndMeth              "Floor"
  SaturateOnIntegerOverflow on
}
Block {
  BlockType           Mux
  Inputs              "4"
  DisplayOption       "none"
  BusObject           "BusObject"
  NonVirtualBus       off
}
Block {
  BlockType           Outport
  Port                "1"
  BusObject           "BusObject"
  BusOutputAsStruct   off
  PortDimensions      "-1"
  SampleTime          "-1"
  DataType            "auto"
  OutDataType          "sfix(16)"
  OutScaling           "2^0"
  SignalType          "auto"
  SamplingMode         "auto"
  OutputWhenDisabled  "held"
  InitialOutput       "[]"
}
Block {
  BlockType           Product
  Inputs              "2"
  Multiplication       "Element-wise(.*)"
  InputSameDT         on
  OutDataTypeMode      "Same as first input"
  OutDataType          "sfix(16)"
  OutScaling           "2^0"
  LockScale            off
  RndMeth              "Zero"
  SaturateOnIntegerOverflow on
  SampleTime          "-1"
}

```

```

Block {
    BlockType          RateTransition
    Integrity           on
    Deterministic       on
    X0                  "0"
    OutPortSampleTime   "-1"
}
Block {
    BlockType          Saturate
    UpperLimit          "0.5"
    LowerLimit          "-0.5"
    LinearizeAsGain     on
    ZeroCross           on
    SampleTime          "-1"
}
Block {
    BlockType          Scope
    Floating            off
    ModelBased          off
    TickLabels          "OneTimeTick"
    ZoomMode            "on"
    Grid                "on"
    TimeRange           "auto"
    YMin                "-5"
    YMax                "5"
    SaveToWorkspace     off
    SaveName             "ScopeData"
    LimitDataPoints     on
    MaxDataPoints        "5000"
    Decimation           "1"
    SampleInput         off
    SampleTime          "0"
}
Block {
    BlockType          "S-Function"
    FunctionName        "system"
    SFunctionModules    "'''"
    PortCounts          "[]"
}
Block {
    BlockType          Signum
    ZeroCross           on
    SampleTime          "-1"
}
Block {
    BlockType          Step
    Time                "1"
    Before              "0"
    After               "1"
    SampleTime          "-1"
    VectorParams1D      on
    ZeroCross           on
}
Block {
    BlockType          SubSystem
    ShowPortLabels      on
    Permissions          "ReadWrite"
}

```

```

    PermitHierarchicalResolution "All"
    SystemSampleTime            "-1"
    RTWFcnNameOpts              "Auto"
    RTWFileNameOpts             "Auto"
    SimViewingDevice            off
    DataTypeOverride             "UseLocalSettings"
    MinMaxOverflowLogging       "UseLocalSettings"
}
Block {
    BlockType                    Sum
    IconShape                    "rectangular"
    Inputs                      "++"
    InputSameDT                  on
    OutDataTypeMode              "Same as first input"
    OutDataType                  "sfix(16)"
    OutScaling                   "2^0"
    LockScale                    off
    RndMeth                      "Floor"
    SaturateOnIntegerOverflow    on
    SampleTime                   "-1"
}
Block {
    BlockType                    Switch
    Criteria                     "u2 >= Threshold"
    Threshold                    "0"
    InputSameDT                  on
    OutDataTypeMode              "Inherit via internal rule"
    RndMeth                      "Floor"
    SaturateOnIntegerOverflow    on
    ZeroCross                    on
    SampleTime                   "-1"
}
Block {
    BlockType                    ToWorkspace
    VariableName                 "simulink_output"
    MaxDataPoints                "1000"
    Decimation                   "1"
    SampleTime                   "0"
    FixptAsFi                    off
}
Block {
    BlockType                    Trigonometry
    Operator                     "sin"
    OutputSignalType             "auto"
    SampleTime                   "-1"
}
Block {
    BlockType                    UniformRandomNumber
    Minimum                     "-1"
    Maximum                     "1"
    Seed                         "0"
    SampleTime                   "-1"
    VectorParams1D               on
}
}
AnnotationDefaults {
    HorizontalAlignment          "center"

```

```

        VerticalAlignment      "middle"
        ForegroundColor        "black"
        BackgroundColor        "white"
        DropShadow              off
        FontName                 "Helvetica"
        FontSize                 10
        FontWeight               "normal"
        FontAngle                "normal"
    }
    LineDefaults {
        FontName                 "Helvetica"
        FontSize                 9
        FontWeight               "normal"
        FontAngle                "normal"
    }
    System {
        Name                     "telescope_model_full"
        Location                  [216, 72, 1438, 849]
        Open                      on
        ModelBrowserVisibility    on
        ModelBrowserWidth         214
        ScreenColor               "white"
        PaperOrientation           "landscape"
        PaperPositionMode         "auto"
        PaperType                  "usletter"
        PaperUnits                 "inches"
        ZoomFactor                 "100"
        ReportName                 "simulink-default.rpt"
        Block {
            BlockType              Constant
            Name                    "0Input"
            Position                [405, 155, 435, 185]
            Value                    "0"
        }
        Block {
            BlockType              Constant
            Name                    "0Input1"
            Position                [25, 295, 55, 325]
            Value                    "pi/12"
        }
        Block {
            BlockType              Constant
            Name                    "0Input2"
            Position                [385, 430, 415, 460]
            Value                    "0"
        }
        Block {
            BlockType              Constant
            Name                    "0Input3"
            Position                [20, 50, 50, 80]
            Value                    "pi/12"
        }
        Block {
            BlockType              Reference
            Name                    "Altitude\nInput\nSwitch"
            Ports                  [2, 1]
            Position                [440,          347, 470, 383]

```



```

SourceBlock      "simulink/Signal\nRouting/Manual Switch"
SourceType       "Manual Switch"
ShowPortLabels   on
sw               "1"
action           "0"
}
Block {
  BlockType       Reference
  Name            "Altitude\nInput\nSwitch1"
  Ports           [2, 1]
  Position        [95, 337, 125, 373]
  NamePlacement   "alternate"
  SourceBlock     "simulink/Signal\nRouting/Manual Switch"
  SourceType      "Manual Switch"
  ShowPortLabels  on
  sw              "0"
  action          "0"
}
Block {
  BlockType       Step
  Name            "Altitude_Input"
  Position        [25, 350, 55, 380]
  Time            "Altitude_stepTime"
  Before          "Altitude_initial"
  After           "Altitude_final"
  SampleTime      "0"
}
Block {
  BlockType       SubSystem
  Name            "Altitude_PID_Controller"
  Ports           [2, 1]
  Position        [155, 343, 320, 387]
  TreatAsAtomicUnit off
  MinAlgLoopOccurrences off
  RTWSystemCode   "Auto"
  System {
    Name          "Altitude_PID_Controller"
    Location       [251, 138, 1179, 773]
    Open           off
    ModelBrowserVisibility on
    ModelBrowserWidth 200
    ScreenColor     "white"
    PaperOrientation "landscape"
    PaperPositionMode "auto"
    PaperType       "usletter"
    PaperUnits      "inches"
    ZoomFactor      "100"
    Block {
      BlockType     Inport
      Name          "Control_Input"
      Position       [40, 133, 70, 147]
      Port           "1"
      IconDisplay    "Port number"
      LatchInput     off
    }
  }
  Block {
    BlockType       Inport

```

```

    Name            "Feedback"
    Position         [40, 208, 70, 222]
    Port             "2"
    IconDisplay      "Port number"
    LatchInput       off
}
Block {
    BlockType        Sum
    Name             "Add"
    Ports            [3, 1]
    Position         [655, 107, 685, 173]
    Inputs           "+++"
    InputSameDT      off
    OutDataTypeMode  "Inherit via internal rule"
    OutScaling       "2^-10"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType        Scope
    Name             "Control Signals"
    Ports            [4]
    Position         [830, 194, 880, 316]
    Location         [1, 46, 1441, 869]
    Open             off
    NumInputPorts    "4"
    List {
        ListType      AxesTitles
        axes1          "%<SignalLabel>"
        axes2          "%<SignalLabel>"
        axes3          "%<SignalLabel>"
        axes4          "%<SignalLabel>"
    }
    YMin             "-5~-5~-5~-5"
    YMax             "5~5~5~5"
    SaveName         "ScopeData1"
    DataFormat       "StructureWithTime"
    LimitDataPoints  off
}
Block {
    BlockType        Derivative
    Name             "Derivative"
    Position         [380, 125, 410, 155]
}
Block {
    BlockType        Gain
    Name             "Derivative_Gain"
    Position         [290, 123, 335, 157]
    Gain             "Kd_a"
    ParameterDataTypeMode "Inherit via internal rule"
    ParameterDataType  "fixdt(1, 16)"
    OutDataTypeMode    "Inherit via internal rule"
    OutDataType        "fixdt(1, 16)"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType        Gain
    Name             "Gain"

```

```

    Position      [115, 200, 145, 230]
    Gain          "-1"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "Integral_Gain"
    Position       [290, 186, 335, 224]
    Gain           "Ki_a"
    ParameterDataTypeMode "Inherit via internal rule"
    ParameterDataType "fixdt(1, 16)"
    OutDataTypeMode "Inherit via internal rule"
    OutDataType     "fixdt(1, 16)"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Integrator
    Name           "Integrator"
    Ports          [1, 1]
    Position       [380, 190, 410, 220]
    IgnoreLimit    off
}
Block {
    BlockType      Gain
    Name           "Proportional_Gain"
    Position       [290, 58, 340, 92]
    Gain           "Kp_a"
    ParameterDataTypeMode "Inherit via internal rule"
    ParameterDataType "fixdt(1, 16)"
    OutDataTypeMode "Inherit via internal rule"
    OutDataType     "fixdt(1, 16)"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Saturate
    Name           "Saturation"
    Position       [750, 125, 780, 155]
    UpperLimit     "12"
    LowerLimit     "-12"
}
Block {
    BlockType      Sum
    Name           "Sum"
    Ports          [2, 1]
    Position       [170, 120, 210, 160]
    ShowName       off
    IconShape      "round"
    Inputs         "|++"
    InputSameDT    off
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Outport
    Name           "Control_Signal"

```

```

    Position      [855, 133, 885, 147]
    IconDisplay   "Port number"
    BusOutputAsStruct off
}
Line {
    SrcBlock      "Integral_Gain"
    SrcPort       1
    DstBlock      "Integrator"
    DstPort       1
}
Line {
    SrcBlock      "Derivative_Gain"
    SrcPort       1
    DstBlock      "Derivative"
    DstPort       1
}
Line {
    SrcBlock      "Proportional_Gain"
    SrcPort       1
    Points        [220, 0]
    Branch {
        Points    [50, 0; 0, 45]
        DstBlock  "Add"
        DstPort   1
    }
    Branch {
        Points    [0, 165]
        DstBlock  "Control Signals"
        DstPort   2
    }
}
Line {
    SrcBlock      "Integrator"
    SrcPort       1
    Points        [180, 0]
    Branch {
        Points    [20, 0; 0, -45]
        DstBlock  "Add"
        DstPort   3
    }
    Branch {
        Points    [0, 95]
        DstBlock  "Control Signals"
        DstPort   4
    }
}
Line {
    SrcBlock      "Sum"
    SrcPort       1
    Points        [15, 0]
    Branch {
        DstBlock  "Derivative_Gain"
        DstPort   1
    }
    Branch {
        Points    [0, -65]
        DstBlock  "Proportional_Gain"
    }
}

```

```

        DstPort          1
    }
    Branch {
        Points           [0, 65]
        DstBlock         "Integral_Gain"
        DstPort          1
    }
}
Line {
    SrcBlock             "Feedback"
    SrcPort              1
    DstBlock             "Gain"
    DstPort              1
}
Line {
    SrcBlock             "Control_Input"
    SrcPort              1
    DstBlock             "Sum"
    DstPort              1
}
Line {
    SrcBlock             "Gain"
    SrcPort              1
    Points               [40, 0]
    DstBlock             "Sum"
    DstPort              2
}
Line {
    SrcBlock             "Add"
    SrcPort              1
    DstBlock             "Saturation"
    DstPort              1
}
Line {
    SrcBlock             "Saturation"
    SrcPort              1
    Points               [25, 0]
    Branch {
        DstBlock         "Control_Signal"
        DstPort          1
    }
    Branch {
        Points           [0, 70]
        DstBlock         "Control Signals"
        DstPort          1
    }
}
Line {
    SrcBlock             "Derivative"
    SrcPort              1
    Points               [165, 0]
    Branch {
        DstBlock         "Add"
        DstPort          2
    }
    Branch {
        Points           [0, 130]

```

```

        DstBlock          "Control Signals"
        DstPort            3
    }
}
Annotation {
    Name          "Altitude_PID_Controller\n"
    Position      [123, 40]
    FontName      "Arial"
    FontSize      20
    FontWeight    "bold"
}
}
}
Block {
    BlockType      SubSystem
    Name          "Altitude_Telescope_Structure"
    Ports         [2, 3]
    Position      [600, 331, 790, 469]
    TreatAsAtomicUnit    off
    MinAlgLoopOccurrences    off
    RTWSystemCode    "Auto"
    System {
        Name          "Altitude_Telescope_Structure"
        Location      [358, 164, 1394, 776]
        Open          off
        ModelBrowserVisibility    on
        ModelBrowserWidth    200
        ScreenColor    "white"
        PaperOrientation    "landscape"
        PaperPositionMode    "auto"
        PaperType      "usletter"
        PaperUnits      "inches"
        ZoomFactor      "100"
        Block {
            BlockType      Inport
            Name          "Control_Input(V_a)"
            Position      [40, 268, 70, 282]
            Port          "1"
            IconDisplay    "Port number"
            LatchInput    off
        }
        Block {
            BlockType      Inport
            Name          "Disturbance(Ts_a)"
            Position      [40, 133, 70, 147]
            Port          "2"
            IconDisplay    "Port number"
            LatchInput    off
        }
        Block {
            BlockType      Sum
            Name          "Add"
            Ports         [3, 1]
            Position      [295, 237, 325, 313]
            Inputs        "+++"
            InputSameDT    off
            OutDataTypeMode    "Inherit via internal rule"

```

```

    OutScaling          "2^-10"
    SaturateOnIntegerOverflow off
    Port {
        PortNumber      1
        Name             "dX/dt"
        RTWStorageClass  "Auto"
        DataLoggingNameMode "SignalName"
        ShowSigGenPortName on
    }
}
Block {
    BlockType      Sum
    Name           "Add1"
    Ports          [2, 1]
    Position       [590, 267, 620, 298]
    InputSameDT    off
    OutDataTypeMode "Inherit via internal rule"
    OutScaling     "2^-10"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "C"
    Position       [485, 253, 545, 297]
    Gain           "C_a"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "D"
    Position       [340, 418, 400, 472]
    Gain           "D_a"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Demux
    Name           "Demux"
    Ports          [1, 3]
    Position       [690, 230, 695, 340]
    BackgroundColor "black"
    ShowName       off
    Outputs        "3"
    DisplayOption   "bar"
}
Block {
    BlockType      Gain
    Name           "Disturbance_Gain_E"
    Position       [175, 117, 225, 163]
    Gain           "E_a"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"

```

```

    OutDataTypeMode      "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "Input_Gain_B"
    Position       [180, 252, 230, 298]
    Gain           "B_a"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Integrator
    Name           "Integrator"
    Ports          [1, 1]
    Position       [375, 260, 405, 290]
    InitialCondition "X_a_ini"
    IgnoreLimit    off
    Port {
        PortNumber      1
        Name             "X"
        RTWStorageClass  "Auto"
        DataLoggingNameMode "SignalName"
        ShowSigGenPortName on
    }
}
Block {
    BlockType      Gain
    Name           "System_Gain_A"
    Position       [335, 334, 395, 386]
    Orientation     "left"
    Gain           "A_a"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      ToWorkspace
    Name           "To Workspace"
    Position       [555, 155, 670, 185]
    VariableName    "Altitude_System_State"
    MaxDataPoints   "inf"
    SampleTime      "-1"
    SaveFormat      "Array"
}
Block {
    BlockType      Outport
    Name           "Theta(Theta_J_a)"
    Position       [785, 243, 815, 257]
    IconDisplay     "Port number"
    BusOutputAsStruct off
}
Block {
    BlockType      Outport

```



```

Name          "Omega(Theta_dot_J_a)"
Position      [785, 278, 815, 292]
Port          "2"
IconDisplay   "Port number"
BusOutputAsStruct off
}
Block {
  BlockType    Outport
  Name         "X_alt(m)"
  Position     [785, 313, 815, 327]
  Port         "3"
  IconDisplay   "Port number"
  BusOutputAsStruct off
}
Line {
  Name         "X"
  Labels       [0, 1]
  SrcBlock     "Integrator"
  SrcPort      1
  Points       [30, 0]
  Branch {
    DstBlock    "C"
    DstPort     1
  }
  Branch {
    Points      [0, -105]
    DstBlock    "To Workspace"
    DstPort     1
  }
  Branch {
    Points      [0, 85]
    DstBlock    "System_Gain_A"
    DstPort     1
  }
}
Line {
  Name         "dX/dt"
  Labels       [0, 1]
  SrcBlock     "Add"
  SrcPort      1
  DstBlock     "Integrator"
  DstPort      1
}
Line {
  SrcBlock     "Input_Gain_B"
  SrcPort      1
  DstBlock     "Add"
  DstPort      2
}
Line {
  SrcBlock     "C"
  SrcPort      1
  DstBlock     "Add1"
  DstPort      1
}
Line {
  SrcBlock     "D"

```

```

        SrcPort      1
        Points       [160, 0; 0, -155]
        DstBlock     "Add1"
        DstPort      2
    }
    Line {
        SrcBlock     "Add1"
        SrcPort      1
        DstBlock     "Demux"
        DstPort      1
    }
    Line {
        SrcBlock     "Control_Input(V_a)"
        SrcPort      1
        Points       [35, 0]
        Branch {
            DstBlock     "Input_Gain_B"
            DstPort      1
        }
        Branch {
            Points       [0, 170]
            DstBlock     "D"
            DstPort      1
        }
    }
    Line {
        SrcBlock     "Demux"
        SrcPort      1
        DstBlock     "Theta(Theta_J_a)"
        DstPort      1
    }
    Line {
        SrcBlock     "Demux"
        SrcPort      2
        DstBlock     "Omega(Theta_dot_J_a)"
        DstPort      1
    }
    Line {
        SrcBlock     "Demux"
        SrcPort      3
        DstBlock     "X_alt(m)"
        DstPort      1
    }
    Line {
        SrcBlock     "Disturbance(Ts_a)"
        SrcPort      1
        DstBlock     "Disturbance_Gain_E"
        DstPort      1
    }
    Line {
        SrcBlock     "System_Gain_A"
        SrcPort      1
        Points       [-60, 0; 0, -60]
        DstBlock     "Add"
        DstPort      3
    }
    Line {

```

```

        SrcBlock      "Disturbance_Gain_E"
        SrcPort       1
        Points        [25, 0; 0, 110]
        DstBlock      "Add"
        DstPort       1
    }
    Annotation {
        Name           "Telescope Altitude Structure Model"
        Position       [199, 46]
        FontName       "Arial"
        FontSize       20
        FontWeight     "bold"
    }
    Annotation {
        Name           "This is the Sate Space Model of telescope a"
        "ltitude motion."
        Position       [38, 71]
        HorizontalAlignment "left"
    }
}
}
Block {
    BlockType      Reference
    Name           "Azimuth\nInput\nSwitch"
    Ports          [2, 1]
    Position       [475, 97, 505, 133]
    SourceBlock    "simulink/Signal\nRouting/Manual Switch"
    SourceType     "Manual Switch"
    ShowPortLabels on
    sw             "1"
    action         "0"
}
Block {
    BlockType      Reference
    Name           "Azimuth\nInput\nSwitch1"
    Ports          [2, 1]
    Position       [80, 87, 110, 123]
    NamePlacement  "alternate"
    SourceBlock    "simulink/Signal\nRouting/Manual Switch"
    SourceType     "Manual Switch"
    ShowPortLabels on
    sw             "0"
    action         "0"
}
Block {
    BlockType      Step
    Name           "Azimuth_Input"
    Position       [15, 100, 45, 130]
    Time           "Azimuth_stepTime"
    Before         "Azimuth_initial"
    After          "Azimuth_final"
    SampleTime     "0"
}
Block {
    BlockType      SubSystem
    Name           "Azimuth_PID_Controller"
    Ports          [2, 1]

```

```

    Position          [160, 93, 325, 137]
    TreatAsAtomicUnit    off
    MinAlgLoopOccurrences off
    RTWSystemCode      "Auto"
    System {
Name          "Azimuth_PID_Controller"
Location      [251, 138, 1179, 773]
Open          off
ModelBrowserVisibility on
ModelBrowserWidth 200
ScreenColor    "white"
PaperOrientation "landscape"
PaperPositionMode "auto"
PaperType      "usletter"
PaperUnits     "inches"
ZoomFactor     "100"
Block {
    BlockType      Inport
    Name           "Control_Input"
    Position       [40, 133, 70, 147]
    Port           "1"
    IconDisplay    "Port number"
    LatchInput     off
}
Block {
    BlockType      Inport
    Name           "Feedback"
    Position       [40, 208, 70, 222]
    Port           "2"
    IconDisplay    "Port number"
    LatchInput     off
}
Block {
    BlockType      Sum
    Name           "Add"
    Ports          [3, 1]
    Position       [655, 107, 685, 173]
    Inputs         "+++"
    InputSameDT    off
    OutDataTypeMode "Inherit via internal rule"
    OutScaling     "2^-10"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Scope
    Name           "Control Signals"
    Ports          [4]
    Position       [830, 194, 880, 316]
    Location       [5, 54, 1445, 877]
    Open          off
    NumInputPorts  "4"
    List {
        ListType      AxesTitles
        axes1          "%<SignalLabel>"
        axes2          "%<SignalLabel>"
        axes3          "%<SignalLabel>"
        axes4          "%<SignalLabel>"
    }
}

```

```

    }
    YMin          "-5~-5~-5~-5"
    YMax          "5~5~5~5"
    SaveName      "ScopeData1"
    DataFormat    "StructureWithTime"
    LimitDataPoints off
}
Block {
    BlockType      Derivative
    Name           "Derivative"
    Position       [380, 125, 410, 155]
}
Block {
    BlockType      Gain
    Name           "Derivative_Gain"
    Position       [290, 125, 320, 155]
    Gain           "Kd"
    ParameterDataTypeMode "Inherit via internal rule"
    ParameterDataType  "fixdt(1, 16)"
    OutDataTypeMode    "Inherit via internal rule"
    OutDataType        "fixdt(1, 16)"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "Gain"
    Position       [115, 200, 145, 230]
    Gain           "-1"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode    "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "Integral_Gain"
    Position       [295, 190, 325, 220]
    Gain           "Ki"
    ParameterDataTypeMode "Inherit via internal rule"
    ParameterDataType  "fixdt(1, 16)"
    OutDataTypeMode    "Inherit via internal rule"
    OutDataType        "fixdt(1, 16)"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Integrator
    Name           "Integrator"
    Ports          [1, 1]
    Position       [380, 190, 410, 220]
    IgnoreLimit    off
}
Block {
    BlockType      Gain
    Name           "Proportional_Gain"
    Position       [290, 60, 320, 90]
    Gain           "Kp"
    ParameterDataTypeMode "Inherit via internal rule"
    ParameterDataType  "fixdt(1, 16)"

```

```

    OutDataTypeMode      "Inherit via internal rule"
    OutDataType           "fixdt(1, 16)"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Saturate
    Name           "Saturation"
    Position       [750, 125, 780, 155]
    UpperLimit     "12"
    LowerLimit     "-12"
}
Block {
    BlockType      Scope
    Name           "Scope"
    Ports          [1]
    Position       [270, 269, 300, 301]
    Location       [600, 507, 924, 746]
    Open           off
    NumInputPorts  "1"
    List {
        ListType      AxesTitles
        axes1          "%<SignalLabel>"
    }
    DataFormat      "StructureWithTime"
    LimitDataPoints off
}
Block {
    BlockType      Sum
    Name           "Sum"
    Ports          [2, 1]
    Position       [170, 120, 210, 160]
    ShowName       off
    IconShape      "round"
    Inputs         " | ++"
    InputSameDT    off
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Outport
    Name           "Control_Signal"
    Position       [855, 133, 885, 147]
    IconDisplay     "Port number"
    BusOutputAsStruct off
}
Line {
    SrcBlock      "Integral_Gain"
    SrcPort       1
    DstBlock      "Integrator"
    DstPort       1
}
Line {
    SrcBlock      "Derivative_Gain"
    SrcPort       1
    DstBlock      "Derivative"
    DstPort       1
}

```

```

Line {
  SrcBlock      "Proportional_Gain"
  SrcPort       1
  Points        [240, 0]
  Branch {
    Points      [50, 0; 0, 45]
    DstBlock    "Add"
    DstPort     1
  }
  Branch {
    Points      [0, 165]
    DstBlock    "Control Signals"
    DstPort     2
  }
}
Line {
  SrcBlock      "Integrator"
  SrcPort       1
  Points        [180, 0]
  Branch {
    Points      [20, 0; 0, -45]
    DstBlock    "Add"
    DstPort     3
  }
  Branch {
    Points      [0, 95]
    DstBlock    "Control Signals"
    DstPort     4
  }
}
Line {
  SrcBlock      "Sum"
  SrcPort       1
  Points        [5, 0]
  Branch {
    Points      [10, 0]
    Branch {
      DstBlock  "Derivative_Gain"
      DstPort   1
    }
    Branch {
      Points    [0, -65]
      DstBlock  "Proportional_Gain"
      DstPort   1
    }
    Branch {
      Points    [0, 65]
      DstBlock  "Integral_Gain"
      DstPort   1
    }
  }
}
Branch {
  Points      [0, 145]
  DstBlock    "Scope"
  DstPort     1
}
}

```

```

Line {
    SrcBlock      "Feedback"
    SrcPort       1
    DstBlock      "Gain"
    DstPort       1
}
Line {
    SrcBlock      "Control_Input"
    SrcPort       1
    DstBlock      "Sum"
    DstPort       1
}
Line {
    SrcBlock      "Gain"
    SrcPort       1
    Points        [40, 0]
    DstBlock      "Sum"
    DstPort       2
}
Line {
    SrcBlock      "Add"
    SrcPort       1
    DstBlock      "Saturation"
    DstPort       1
}
Line {
    SrcBlock      "Saturation"
    SrcPort       1
    Points        [25, 0]
    Branch {
        DstBlock      "Control_Signal"
        DstPort       1
    }
    Branch {
        Points        [0, 70]
        DstBlock      "Control Signals"
        DstPort       1
    }
}
Line {
    SrcBlock      "Derivative"
    SrcPort       1
    Points        [165, 0]
    Branch {
        DstBlock      "Add"
        DstPort       2
    }
    Branch {
        Points        [0, 130]
        DstBlock      "Control Signals"
        DstPort       3
    }
}
Annotation {
    Name          "Azimuth PID Controller\n"
    Position      [128, 34]
    FontName      "Arial"
}

```



```

        FontSize          20
        FontWeight        "bold"
    }
}
}
Block {
    BlockType          SubSystem
    Name               "Azimuth_Telescope_Structure"
    Ports              [2, 3]
    Position            [595, 78, 785, 222]
    TreatAsAtomicUnit   off
    MinAlgLoopOccurrences off
    RTWSystemCode       "Auto"
    System {
        Name           "Azimuth_Telescope_Structure"
        Location        [341, 81, 1377, 693]
        Open            off
        ModelBrowserVisibility on
        ModelBrowserWidth 200
        ScreenColor      "white"
        PaperOrientation  "landscape"
        PaperPositionMode "auto"
        PaperType         "usletter"
        PaperUnits        "inches"
        ZoomFactor        "100"
    }
    Block {
        BlockType      Inport
        Name           "Control_Input(V)"
        Position        [40, 268, 70, 282]
        Port            "1"
        IconDisplay     "Port number"
        LatchInput      off
    }
    Block {
        BlockType      Inport
        Name           "Disturbance(Ts)"
        Position        [40, 133, 70, 147]
        Port            "2"
        IconDisplay     "Port number"
        LatchInput      off
    }
    Block {
        BlockType      Sum
        Name           "Add"
        Ports          [3, 1]
        Position        [295, 237, 325, 313]
        Inputs          "+++"
        InputSameDT     off
        OutDataTypeMode "Inherit via internal rule"
        OutScaling      "2^-10"
        SaturateOnIntegerOverflow off
        Port {
            PortNumber    1
            Name          "dX/dt"
            RTWStorageClass "Auto"
            DataLoggingNameMode "SignalName"
            ShowSigGenPortName on
        }
    }
}

```

```

    }
}
Block {
    BlockType      Sum
    Name           "Add1"
    Ports          [2, 1]
    Position       [590, 267, 620, 298]
    InputSameDT    off
    OutDataTypeMode "Inherit via internal rule"
    OutScaling     "2^-10"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "C"
    Position       [485, 253, 545, 297]
    Gain           "C"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "D"
    Position       [340, 418, 400, 472]
    Gain           "D"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Demux
    Name           "Demux"
    Ports          [1, 3]
    Position       [690, 230, 695, 340]
    BackgroundColor "black"
    ShowName       off
    Outputs        "3"
    DisplayOption   "bar"
}
Block {
    BlockType      Gain
    Name           "Disturbance_Gain_E"
    Position       [175, 117, 225, 163]
    Gain           "E"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "Input_Gain_B"
    Position       [180, 252, 230, 298]
    Gain           "B"

```

```

    Multiplication      "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode     "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Integrator
    Name           "Integrator"
    Ports          [1, 1]
    Position       [375, 260, 405, 290]
    InitialCondition "X_ini"
    IgnoreLimit    off
    Port {
        PortNumber      1
        Name             "X"
        RTWStorageClass  "Auto"
        DataLoggingNameMode "SignalName"
        ShowSigGenPortName on
    }
}
Block {
    BlockType      Gain
    Name           "System_Gain_A"
    Position       [335, 334, 395, 386]
    Orientation    "left"
    Gain           "A"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode     "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      ToWorkspace
    Name           "To Workspace"
    Position       [555, 155, 670, 185]
    VariableName    "Azimuth_System_State"
    MaxDataPoints   "inf"
    SampleTime      "-1"
    SaveFormat      "Array"
}
Block {
    BlockType      Outport
    Name           "Theta(Theta_J)"
    Position       [785, 243, 815, 257]
    IconDisplay    "Port number"
    BusOutputAsStruct off
}
Block {
    BlockType      Outport
    Name           "Omega(theta_dot_J)"
    Position       [785, 278, 815, 292]
    Port           "2"
    IconDisplay    "Port number"
    BusOutputAsStruct off
}
Block {
    BlockType      Outport

```

```

Name          "X_azi(m)"
Position      [785, 313, 815, 327]
Port          "3"
IconDisplay   "Port number"
BusOutputAsStruct off
}
Line {
  Name          "X"
  Labels        [0, 1]
  SrcBlock      "Integrator"
  SrcPort       1
  Points        [30, 0]
  Branch {
    DstBlock     "C"
    DstPort      1
  }
  Branch {
    Points       [0, -105]
    DstBlock     "To Workspace"
    DstPort      1
  }
  Branch {
    Points       [0, 85]
    DstBlock     "System_Gain_A"
    DstPort      1
  }
}
Line {
  Name          "dx/dt"
  Labels        [0, 1]
  SrcBlock      "Add"
  SrcPort       1
  DstBlock      "Integrator"
  DstPort       1
}
Line {
  SrcBlock      "Input_Gain_B"
  SrcPort       1
  DstBlock      "Add"
  DstPort       2
}
Line {
  SrcBlock      "C"
  SrcPort       1
  DstBlock      "Add1"
  DstPort       1
}
Line {
  SrcBlock      "D"
  SrcPort       1
  Points        [160, 0; 0, -155]
  DstBlock      "Add1"
  DstPort       2
}
Line {
  SrcBlock      "Add1"
  SrcPort       1

```

```

        DstBlock      "Demux"
        DstPort      1
    }
    Line {
        SrcBlock      "Control_Input(V)"
        SrcPort      1
        Points        [35, 0]
        Branch {
            DstBlock      "Input_Gain_B"
            DstPort      1
        }
        Branch {
            Points        [0, 170]
            DstBlock      "D"
            DstPort      1
        }
    }
    Line {
        SrcBlock      "Demux"
        SrcPort      1
        DstBlock      "Theta(Theta_J)"
        DstPort      1
    }
    Line {
        SrcBlock      "Demux"
        SrcPort      2
        DstBlock      "Omega(theta_dot_J)"
        DstPort      1
    }
    Line {
        SrcBlock      "Demux"
        SrcPort      3
        DstBlock      "X_azi(m)"
        DstPort      1
    }
    Line {
        SrcBlock      "Disturbance(Ts)"
        SrcPort      1
        DstBlock      "Disturbance_Gain_E"
        DstPort      1
    }
    Line {
        SrcBlock      "System_Gain_A"
        SrcPort      1
        Points        [-60, 0; 0, -60]
        DstBlock      "Add"
        DstPort      3
    }
    Line {
        SrcBlock      "Disturbance_Gain_E"
        SrcPort      1
        Points        [25, 0; 0, 110]
        DstBlock      "Add"
        DstPort      1
    }
    Annotation {
        Name          "Telescope      Azimuth Structure Model"
    }

```

```

        Position      [204, 31]
        FontName      "Arial"
        FontSize      20
        FontWeight    "bold"
    }
    Annotation {
        Name          "This is the Sate Space Model of telescope a"
"zimuth motion."
        Position      [42, 58]
        HorizontalAlignment "left"
    }
}
}
Block {
    BlockType        Clock
    Name             "Clock"
    Position          [295, 40, 315, 60]
    Decimation        "10"
}
Block {
    BlockType        Constant
    Name             "Constant"
    Position          [175, 685, 205, 715]
    Value            "0"
}
Block {
    BlockType        SubSystem
    Name             "Disturbance_Input"
    Ports            [4, 2]
    Position          [550, 524, 790, 646]
    Orientation       "left"
    TreatAsAtomicUnit off
    MinAlgLoopOccurrences off
    RTWSystemCode     "Auto"
    System {
        Name          "Disturbance_Input"
        Location       [209, 80, 1392, 701]
        Open           off
        ModelBrowserVisibility on
        ModelBrowserWidth 220
        ScreenColor    "white"
        PaperOrientation "landscape"
        PaperPositionMode "auto"
        PaperType       "usletter"
        PaperUnits       "inches"
        ZoomFactor       "100"
        Block {
            BlockType    Inport
            Name          "Altitude_\nAngular_Position\n(Theta_J_a)(ra"
"dius)"
            Position      [120, 143, 150, 157]
            Port          "1"
            IconDisplay    "Port number"
            LatchInput     off
        }
        Block {
            BlockType    Inport

```

```

        Name          "Azimuth_Angular_Position\n(Theta_J)(radius)"
        Position       [90, 703, 120, 717]
        Port           "2"
        IconDisplay     "Port number"
        LatchInput      off
    }
    Block {
        BlockType       Inport
        Name             "Wind_Direction\n(radius)"
        Position         [115, 353, 145, 367]
        Port             "3"
        IconDisplay      "Port number"
        LatchInput       off
    }
    Block {
        BlockType       Inport
        Name             "Wind_Speed\n(m/s)"
        Position         [115, 423, 145, 437]
        Port             "4"
        IconDisplay      "Port number"
        LatchInput       off
    }
    Block {
        BlockType       SubSystem
        Name             "Altitude_Wind_Model"
        Ports            [5, 3]
        Position         [400, 112, 770, 468]
        TreatAsAtomicUnit off
        MinAlgLoopOccurrences off
        RTWSystemCode    "Auto"
        System {
            Name          "Altitude_Wind_Model"
            Location       [209, 80, 1419, 701]
            Open           off
            ModelBrowserVisibility on
            ModelBrowserWidth 200
            ScreenColor     "white"
            PaperOrientation "landscape"
            PaperPositionMode "auto"
            PaperType       "usletter"
            PaperUnits      "inches"
            ZoomFactor      "100"
            Block {
                BlockType       Inport
                Name             "Telescope_Azimuth\nAngular_Position\n(T"
                Position         [80, 133, 110, 147]
                Port             "1"
                IconDisplay      "Port number"
                LatchInput       off
            }
            Block {
                BlockType       Inport
                Name             "OTA_Altitude_Angular_Position\n(Theta_J"
                Position         [80, 208, 110, 222]
                Port             "2"

```

```

        IconDisplay      "Port number"
        LatchInput       off
    }
    Block {
        BlockType         Inport
        Name               "mm_Altitude_Angular_Position\n(Theta_Jm"
        "_a)(radius)"
        Position           [80, 308, 110, 322]
        Port               "3"
        IconDisplay        "Port number"
        LatchInput         off
    }
    Block {
        BlockType         Inport
        Name               "Wind_Direction\n(Theta_w)(radius)"
        Position           [80, 393, 110, 407]
        Port               "4"
        IconDisplay        "Port number"
        LatchInput         off
    }
    Block {
        BlockType         Inport
        Name               "Wind_Speed\n(m/s)"
        Position           [85, 483, 115, 497]
        Port               "5"
        IconDisplay        "Port number"
        LatchInput         off
    }
    Block {
        BlockType         Sum
        Name               "Add"
        Ports              [2, 1]
        Position           [215, 132, 245, 163]
        Inputs              "+- "
        InputSameDT        off
        OutDataTypeMode     "Inherit via internal rule"
        OutScaling          "2^-10"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType         Gain
        Name               "Gain2"
        Position           [695, 127, 935, 183]
        Gain                "0.5*p_air*A_OTA_a*0.5*L_OTA"
        ParameterDataTypeMode "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType         Gain
        Name               "Gain3"
        Position           [700, 282, 940, 338]
        Gain                "-0.5*p_air*A_mm_a*0.5*L_mm"
        ParameterDataTypeMode "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType         Gain

```



```

Name          "Gain4"
Position      [700, 412, 965, 468]
Gain          "0.5*p_air*0.5*L_Fork*A_Fork_a"
ParameterDataTypeMode "Inherit via internal rule"
SaturateOnIntegerOverflow off
}
Block {
  BlockType    Math
  Name         "Math\nFunction"
  Ports        [1, 1]
  Position     [510, 145, 540, 175]
  Operator     "square"
}
Block {
  BlockType    Math
  Name         "Math\nFunction1"
  Ports        [1, 1]
  Position     [515, 300, 545, 330]
  Operator     "square"
}
Block {
  BlockType    Math
  Name         "Math\nFunction2"
  Ports        [1, 1]
  Position     [520, 430, 550, 460]
  Operator     "square"
}
Block {
  BlockType    Product
  Name         "Product"
  Ports        [3, 1]
  Position     [415, 143, 445, 177]
  Inputs       "3"
  InputSameDT  off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType    Product
  Name         "Product1"
  Ports        [3, 1]
  Position     [415, 298, 445, 332]
  Inputs       "3"
  InputSameDT  off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType    Product
  Name         "Product2"
  Ports        [2, 1]
  Position     [415, 426, 445, 459]
  Inputs       "3"
  InputSameDT  off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {

```

```

        BlockType          Product
        Name                "Product3"
        Ports               [2, 1]
        Position            [605, 137, 635, 168]
        InputSameDT         off
        OutDataTypeMode     "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType          Product
        Name                "Product4"
        Ports               [2, 1]
        Position            [615, 292, 645, 323]
        InputSameDT         off
        OutDataTypeMode     "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType          Product
        Name                "Product5"
        Ports               [2, 1]
        Position            [615, 422, 645, 453]
        InputSameDT         off
        OutDataTypeMode     "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType          Signum
        Name                "Sign"
        Position            [510, 75, 540, 105]
    }
    Block {
        BlockType          Signum
        Name                "Sign1"
        Position            [515, 230, 545, 260]
    }
    Block {
        BlockType          Signum
        Name                "Sign2"
        Position            [520, 365, 550, 395]
    }
    Block {
        BlockType          Trigonometry
        Name                "Trigonometric\nFunction"
        Ports               [1, 1]
        Position            [275, 135, 305, 165]
        Operator            "cos"
    }
    Block {
        BlockType          Trigonometry
        Name                "Trigonometric\nFunction1"
        Ports               [1, 1]
        Position            [275, 300, 305, 330]
    }
    Block {
        BlockType          Trigonometry
        Name                "Trigonometric\nFunction2"

```

```

Ports          [1, 1]
Position       [275, 200, 305, 230]
}
Block {
  BlockType      Outport
  Name           "Wind_Torque_OTA\n(Ts1_a)(Nm) "
  Position       [990, 148, 1020, 162]
  IconDisplay    "Port number"
  BusOutputAsStruct off
}
Block {
  BlockType      Outport
  Name           "Wind_Torque_mm\n(Ts2_a)(Nm) "
  Position       [1000, 303, 1030, 317]
  Port           "2"
  IconDisplay    "Port number"
  BusOutputAsStruct off
}
Block {
  BlockType      Outport
  Name           "Wind_Torque_Fork\n(Ts3_a)(Nm) "
  Position       [1005, 433, 1035, 447]
  Port           "3"
  IconDisplay    "Port number"
  BusOutputAsStruct off
}
Line {
  SrcBlock       "Gain3"
  SrcPort        1
  DstBlock       "Wind_Torque_mm\n(Ts2_a)(Nm) "
  DstPort        1
}
Line {
  SrcBlock       "Gain2"
  SrcPort        1
  DstBlock       "Wind_Torque_OTA\n(Ts1_a)(Nm) "
  DstPort        1
}
Line {
  SrcBlock       "Wind_Speed\n(m/s) "
  SrcPort        1
  Points         [265, 0; 0, -40]
  Branch {
    DstBlock      "Product2"
    DstPort       2
  }
  Branch {
    Points        [0, -125]
    Branch {
      Points       [0, -155]
      DstBlock     "Product"
      DstPort      3
    }
  }
  Branch {
    DstBlock      "Product1"
    DstPort       3
  }
}

```

```

    }
  }
  Line {
    SrcBlock      "Gain4"
    SrcPort        1
    DstBlock       "Wind_Torque_Fork\n(Ts3_a)(Nm) "
    DstPort        1
  }
  Line {
    SrcBlock      "OTA_Altitude_Angular_Position\n(Theta_J"
    "O_a)(radius)"
    SrcPort        1
    DstBlock       "Trigonometric\nFunction2"
    DstPort        1
  }
  Line {
    SrcBlock      "Trigonometric\nFunction1"
    SrcPort        1
    DstBlock       "Product1"
    DstPort        2
  }
  Line {
    SrcBlock      "mm_Altitude_Angular_Position\n(Theta_Jm"
    "_a)(radius)"
    SrcPort        1
    DstBlock       "Trigonometric\nFunction1"
    DstPort        1
  }
  Line {
    SrcBlock      "Trigonometric\nFunction2"
    SrcPort        1
    Points         [55, 0; 0, -55]
    DstBlock       "Product"
    DstPort        2
  }
  Line {
    SrcBlock      "Trigonometric\nFunction"
    SrcPort        1
    Points         [0, 0; 25, 0]
    Branch {
      Points       [0, 285]
      DstBlock     "Product2"
      DstPort      1
    }
    Branch {
      Points       [15, 0]
      Branch {
        Points     [0, 155]
        DstBlock   "Product1"
        DstPort    1
      }
    }
    Branch {
      DstBlock     "Product"
      DstPort      1
    }
  }
}

```

```

Line {
  SrcBlock      "Add"
  SrcPort       1
  DstBlock      "Trigonometric\nFunction"
  DstPort       1
}
Line {
  SrcBlock      "Wind_Direction\n(Theta_w)(radius)"
  SrcPort       1
  Points        [65, 0; 0, -245]
  DstBlock      "Add"
  DstPort       2
}
Line {
  SrcBlock      "Telescope_Azimuth\nAngular_Position\n(T"
  SrcPort       1
  DstBlock      "Add"
  DstPort       1
}
Line {
  SrcBlock      "Product"
  SrcPort       1
  Points        [0, 0; 25, 0]
  Branch {
    DstBlock      "Math\nFunction"
    DstPort       1
  }
  Branch {
    Points        [0, -70]
    DstBlock      "Sign"
    DstPort       1
  }
}
Line {
  SrcBlock      "Math\nFunction"
  SrcPort       1
  DstBlock      "Product3"
  DstPort       2
}
Line {
  SrcBlock      "Sign"
  SrcPort       1
  Points        [25, 0; 0, 55]
  DstBlock      "Product3"
  DstPort       1
}
Line {
  SrcBlock      "Product3"
  SrcPort       1
  DstBlock      "Gain2"
  DstPort       1
}
Line {
  SrcBlock      "Product1"
  SrcPort       1
  Points        [0,      0; 30, 0]

```

```

    Branch {
DstBlock      "Math\nFunction1"
DstPort      1
    }
    Branch {
Points        [0, -70]
DstBlock      "Sign1"
DstPort      1
    }
}
Line {
    SrcBlock      "Math\nFunction1"
    SrcPort      1
    DstBlock      "Product4"
    DstPort      2
}
Line {
    SrcBlock      "Sign1"
    SrcPort      1
    Points        [25, 0; 0, 55]
    DstBlock      "Product4"
    DstPort      1
}
Line {
    SrcBlock      "Product4"
    SrcPort      1
    DstBlock      "Gain3"
    DstPort      1
}
Line {
    SrcBlock      "Product2"
    SrcPort      1
    Points        [0, 0; 35, 0]
    Branch {
DstBlock      "Math\nFunction2"
DstPort      1
    }
    Branch {
Points        [0, -65]
DstBlock      "Sign2"
DstPort      1
    }
}
Line {
    SrcBlock      "Math\nFunction2"
    SrcPort      1
    DstBlock      "Product5"
    DstPort      2
}
Line {
    SrcBlock      "Sign2"
    SrcPort      1
    Points        [20, 0; 0, 50]
    DstBlock      "Product5"
    DstPort      1
}
Line {

```

```

        SrcBlock          "Product5"
        SrcPort           1
        DstBlock          "Gain4"
        DstPort           1
    }
    Annotation {
        Name                "Telescope Altitude Wind Model"
        Position            [346, 54]
        FontName            "Arial"
        FontSize            16
        FontWeight          "bold"
    }
    Annotation {
        Name                "mm --> Mirror Mount"
        Position            [284, 73]
    }
}
}
Block {
    BlockType              SubSystem
    Name                   "Azimuth_Wind_Model"
    Ports                  [4, 2]
    Position               [405, 667, 775, 1023]
    TreatAsAtomicUnit      off
    MinAlgLoopOccurrences  off
    RTWSystemCode          "Auto"
    System {
        Name                "Azimuth_Wind_Model"
        Location            [212, 81, 1325, 473]
        Open                off
        ModelBrowserVisibility on
        ModelBrowserWidth   200
        ScreenColor         "white"
        PaperOrientation     "landscape"
        PaperPositionMode   "auto"
        PaperType            "usletter"
        PaperUnits           "inches"
        ZoomFactor          "100"
        Block {
            BlockType        Inport
            Name              "OTA_Altitude_Angular_Position\n(Theta_J"
            "O_a)(radius)"
            Position          [85, 108, 115, 122]
            Port              "1"
            IconDisplay       "Port number"
            LatchInput        off
        }
        Block {
            BlockType        Inport
            Name              "mm_Altitude_Angular_Position\n(Theta_Jm"
            "_a)(radius)"
            Position          [85, 203, 115, 217]
            Port              "2"
            IconDisplay       "Port number"
            LatchInput        off
        }
    }
}
Block {

```

```

        BlockType      Inport
        Name            "Wind_Direction\n(Theta_w)(radius)"
        Position        [85, 293, 115, 307]
        Port            "3"
        IconDisplay      "Port number"
        LatchInput      off
    }
    Block {
        BlockType      Inport
        Name            "Wind_Speed\n(m/s)"
        Position        [85, 348, 115, 362]
        Port            "4"
        IconDisplay      "Port number"
        LatchInput      off
    }
    Block {
        BlockType      Sum
        Name            "Add"
        Ports          [2, 1]
        Position        [210, 107, 240, 138]
        Inputs          "+-"
        InputSameDT      off
        OutDataTypeMode "Inherit via internal rule"
        OutScaling       "2^-10"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType      Sum
        Name            "Add1"
        Ports          [2, 1]
        Position        [210, 202, 240, 233]
        Inputs          "+-"
        InputSameDT      off
        OutDataTypeMode "Inherit via internal rule"
        OutScaling       "2^-10"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType      Gain
        Name            "Gain2"
        Position        [745, 102, 985, 158]
        Gain            "0.5*p_air*A_OTA*0.5*L_OTA"
        ParameterDataTypeMode "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType      Gain
        Name            "Gain3"
        Position        [745, 197, 985, 253]
        Gain            "-0.5*p_air*A_mm*0.5*L_mm"
        ParameterDataTypeMode "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType      Math
        Name            "Math\nFunction"
        Ports          [1, 1]
    }

```



```

        Position          [520, 215, 550, 245]
        Operator          "square"
    }
    Block {
        BlockType          Math
        Name                "Math\nFunction1"
        Ports              [1, 1]
        Position            [480, 120, 510, 150]
        Operator            "square"
    }
    Block {
        BlockType          Product
        Name                "Product"
        Ports              [2, 1]
        Position            [390, 116, 420, 149]
        InputSameDT         off
        OutDataTypeMode     "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType          Product
        Name                "Product1"
        Ports              [2, 1]
        Position            [390, 211, 420, 244]
        InputSameDT         off
        OutDataTypeMode     "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType          Product
        Name                "Product2"
        Ports              [2, 1]
        Position            [570, 112, 600, 143]
        InputSameDT         off
        OutDataTypeMode     "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType          Product
        Name                "Product3"
        Ports              [2, 1]
        Position            [660, 207, 690, 238]
        InputSameDT         off
        OutDataTypeMode     "Inherit via internal rule"
        SaturateOnIntegerOverflow off
    }
    Block {
        BlockType          Signum
        Name                "Sign"
        Position            [485, 55, 515, 85]
    }
    Block {
        BlockType          Signum
        Name                "Sign1"
        Position            [540, 165, 570, 195]
    }
    Block {

```

```

        BlockType      Trigonometry
        Name            "Trigonometric\nFunction"
        Ports          [1, 1]
        Position        [275, 110, 305, 140]
    }
    Block {
        BlockType      Trigonometry
        Name            "Trigonometric\nFunction1"
        Ports          [1, 1]
        Position        [275, 205, 305, 235]
    }
    Block {
        BlockType      Outport
        Name            "Wind_Torque_OTA\n(n(Nm) "
        Position        [1035, 123, 1065, 137]
        IconDisplay     "Port number"
        BusOutputAsStruct  off
    }
    Block {
        BlockType      Outport
        Name            "Wind_Torque_mm\n(n(Nm) "
        Position        [1040, 218, 1070, 232]
        Port            "2"
        IconDisplay     "Port number"
        BusOutputAsStruct  off
    }
    Line {
        SrcBlock        "Add"
        SrcPort          1
        DstBlock        "Trigonometric\nFunction"
        DstPort          1
    }
    Line {
        SrcBlock        "Trigonometric\nFunction"
        SrcPort          1
        DstBlock        "Product"
        DstPort          1
    }
    Line {
        SrcBlock        "Gain2"
        SrcPort          1
        DstBlock        "Wind_Torque_OTA\n(n(Nm) "
        DstPort          1
    }
    Line {
        SrcBlock        "Gain3"
        SrcPort          1
        DstBlock        "Wind_Torque_mm\n(n(Nm) "
        DstPort          1
    }
    Line {
        SrcBlock        "OTA_Altitude_Angular_Position\n(Theta_J"
        SrcPort          1
        DstBlock        "Add"
        DstPort          1
    }
}

```

```

Line {
  SrcBlock      "Wind_Direction\n(Theta_w)(radius)"
  SrcPort      1
  Points       [60, 0; 0, -75]
  Branch {
    Points      [0, -95]
    DstBlock    "Add"
    DstPort     2
  }
  Branch {
    DstBlock    "Add1"
    DstPort     2
  }
}
Line {
  SrcBlock      "Trigonometric\nFunction1"
  SrcPort      1
  DstBlock      "Product1"
  DstPort      1
}
Line {
  SrcBlock      "Add1"
  SrcPort      1
  DstBlock      "Trigonometric\nFunction1"
  DstPort      1
}
Line {
  SrcBlock      "mm_Altitude_Angular_Position\n(Theta_Jm"
  SrcPort      1
  DstBlock      "Add1"
  DstPort      1
}
Line {
  SrcBlock      "Wind_Speed\n(m/s)"
  SrcPort      1
  Points       [230, 0; 0, -120]
  Branch {
    Points      [0, -95]
    DstBlock    "Product"
    DstPort     2
  }
  Branch {
    DstBlock    "Product1"
    DstPort     2
  }
}
Line {
  SrcBlock      "Product"
  SrcPort      1
  Points       [15, 0]
  Branch {
    DstBlock    "Math\nFunction1"
    DstPort     1
  }
  Branch {
    Points      [0, -65]

```

```

DstBlock      "Sign"
DstPort      1
}
}
Line {
    SrcBlock      "Math\nFunction1"
    SrcPort      1
    DstBlock      "Product2"
    DstPort      2
}
Line {
    SrcBlock      "Sign"
    SrcPort      1
    Points      [15, 0; 0, 50]
    DstBlock      "Product2"
    DstPort      1
}
Line {
    SrcBlock      "Product2"
    SrcPort      1
    DstBlock      "Gain2"
    DstPort      1
}
Line {
    SrcBlock      "Product1"
    SrcPort      1
    Points      [20, 0]
    Branch {
        DstBlock      "Math\nFunction"
        DstPort      1
    }
    Branch {
        Points      [0, -50]
        DstBlock      "Sign1"
        DstPort      1
    }
}
Line {
    SrcBlock      "Math\nFunction"
    SrcPort      1
    DstBlock      "Product3"
    DstPort      2
}
Line {
    SrcBlock      "Sign1"
    SrcPort      1
    Points      [35, 0; 0, 35]
    DstBlock      "Product3"
    DstPort      1
}
Line {
    SrcBlock      "Product3"
    SrcPort      1
    DstBlock      "Gain3"
    DstPort      1
}
Annotation {

```

```

        Name          "mm --> Mirror Mount"
        Position      [274, 48]
    }
    Annotation {
        Name          "Telescope Azimuth Wind Model"
        Position      [341, 24]
        FontName      "Arial"
        FontSize      16
        FontWeight    "bold"
    }
}
}
Block {
    BlockType      SubSystem
    Name          "Gravity_Model"
    Ports          [2, 2]
    Position      [400, 491, 845, 624]
    TreatAsAtomicUnit off
    MinAlgLoopOccurrences off
    RTWSystemCode "Auto"
    System {
        Name          "Gravity_Model"
        Location      [695, 84, 1374, 426]
        Open          off
        ModelBrowserVisibility on
        ModelBrowserWidth 200
        ScreenColor    "white"
        PaperOrientation "landscape"
        PaperPositionMode "auto"
        PaperType      "usletter"
        PaperUnits      "inches"
        ZoomFactor      "100"
        Block {
            BlockType      Inport
            Name          "OTA_Altitude_Angular_Position\n(Theta_J"
            "O_a)(radius)"
            Position      [110, 103, 140, 117]
            Port          "1"
            IconDisplay    "Port number"
            LatchInput    off
        }
        Block {
            BlockType      Inport
            Name          "mm_Altitude_Angular_Position\n(Theta_Jm"
            "_a)(radius)"
            Position      [110, 238, 140, 252]
            Port          "2"
            IconDisplay    "Port number"
            LatchInput    off
        }
        Block {
            BlockType      Gain
            Name          "Gain2"
            Position      [295, 89, 480, 131]
            Gain          "-m_OTA*g*0.5*L_OTA"
            ParameterDataTypeMode "Inherit via internal rule"
            OutDataTypeMode "Inherit via internal rule"
        }
    }
}

```

```

    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Gain
    Name           "Gain3"
    Position       [300, 222, 475, 268]
    Gain           "m_mm*g*0.5*L_mm"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode  "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType      Trigonometry
    Name           "Trigonometric\nFunction"
    Ports          [1, 1]
    Position       [225, 95, 255, 125]
    Operator        "cos"
}
Block {
    BlockType      Trigonometry
    Name           "Trigonometric\nFunction1"
    Ports          [1, 1]
    Position       [230, 230, 260, 260]
    Operator        "cos"
}
Block {
    BlockType      Outputport
    Name           "Gravitational_Torque_OTA\n(Ts4_a)(Nm) "
    Position       [530, 103, 560, 117]
    IconDisplay    "Port number"
    BusOutputAsStruct off
}
Block {
    BlockType      Outputport
    Name           "Gravitational_Torque_mm\n(Ts5_a)(Nm) "
    Position       [530, 238, 560, 252]
    Port           "2"
    IconDisplay    "Port number"
    BusOutputAsStruct off
}
Line {
    SrcBlock      "Gain3"
    SrcPort       1
    DstBlock      "Gravitational_Torque_mm\n(Ts5_a)(Nm) "
    DstPort       1
}
Line {
    SrcBlock      "Gain2"
    SrcPort       1
    DstBlock      "Gravitational_Torque_OTA\n(Ts4_a)(Nm) "
    DstPort       1
}
Line {
    SrcBlock      "Trigonometric\nFunction1"
    SrcPort       1
    DstBlock      "Gain3"
    DstPort       1
}

```

```

    }
    Line {
        SrcBlock          "mm_Altitude_Angular_Position\n(Theta_Jm"
        "_a)(radius)"
        SrcPort            1
        DstBlock           "Trigonometric\nFunction1"
        DstPort            1
    }
    Line {
        SrcBlock          "Trigonometric\nFunction"
        SrcPort            1
        DstBlock           "Gain2"
        DstPort            1
    }
    Line {
        SrcBlock          "OTA_Altitude_Angular_Position\n(Theta_J"
        "O_a)(radius)"
        SrcPort            1
        DstBlock           "Trigonometric\nFunction"
        DstPort            1
    }
    Annotation {
        Name               "Telescope Gravity Model"
        Position           [331, 39]
        FontName           "Arial"
        FontSize           16
        FontWeight         "bold"
    }
    Annotation {
        Name               "mm --> Mirror Mount"
        Position           [314, 308]
    }
}
Block {
    BlockType             Mux
    Name                  "Mux"
    Ports                 [2, 1]
    Position               [810, 665, 820, 1025]
    ShowName              off
    Inputs                "2"
    DisplayOption          "bar"
}
Block {
    BlockType             Mux
    Name                  "Mux1"
    Ports                 [5, 1]
    Position               [945, 177, 950, 503]
    ShowName              off
    Inputs                "5"
    DisplayOption          "bar"
}
Block {
    BlockType             Outport
    Name                  "Ts"
    Position               [875, 838, 905, 852]
    IconDisplay            "Port      number"

```

```

        BusOutputAsStruct    off
    }
    Block {
        BlockType            Outport
        Name                  "Ts_a"
        Position              [995, 333, 1025, 347]
        Port                  "2"
        IconDisplay            "Port number"
        BusOutputAsStruct    off
    }
    Line {
        SrcBlock              "Altitude_\nAngular_Position\n(Theta_J_a)(ra"
"dius)"
        SrcPort              1
        Points                [75, 0]
        Branch {
            Points            [130, 0]
            Branch {
                Points        [0, 375]
                DstBlock       "Gravity_Model"
                DstPort        1
            }
            Branch {
                Points        [20, 0]
                Branch {
                    DstBlock   "Altitude_Wind_Model"
                    DstPort    1
                }
                Branch {
                    Points     [0, 70]
                    DstBlock   "Altitude_Wind_Model"
                    DstPort    2
                }
            }
        }
    }
    Branch {
        Points            [0, 140; 105, 0]
        Branch {
            DstBlock       "Altitude_Wind_Model"
            DstPort        3
        }
        Branch {
            Points         [0, 300]
            DstBlock       "Gravity_Model"
            DstPort        2
        }
    }
}
Line {
    SrcBlock              "Azimuth_Angular_Position\n(Theta_J)(radius)"
    SrcPort              1
    Points                [65, 0]
    Branch {
        DstBlock          "Azimuth_Wind_Model"
        DstPort            1
    }
    Branch {

```



```

        Points      [0, 90]
        DstBlock    "Azimuth_Wind_Model"
        DstPort     2
    }
}
Line {
    SrcBlock    "Altitude_Wind_Model"
    SrcPort     1
    Points      [75, 0; 0, 40]
    DstBlock    "Mux1"
    DstPort     1
}
Line {
    SrcBlock    "Gravity_Model"
    SrcPort     1
    Points      [40, 0; 0, -120]
    DstBlock    "Mux1"
    DstPort     4
}
Line {
    SrcBlock    "Gravity_Model"
    SrcPort     2
    Points      [65, 0; 0, -120]
    DstBlock    "Mux1"
    DstPort     5
}
Line {
    SrcBlock    "Wind_Speed\n(m/s)"
    SrcPort     1
    Points      [130, 0]
    Branch {
        Points      [0, 550]
        DstBlock    "Azimuth_Wind_Model"
        DstPort     4
    }
    Branch {
        Points      [0, 0]
        DstBlock    "Altitude_Wind_Model"
        DstPort     5
    }
}
Line {
    SrcBlock    "Mux"
    SrcPort     1
    DstBlock    "Ts"
    DstPort     1
}
Line {
    SrcBlock    "Mux1"
    SrcPort     1
    DstBlock    "Ts_a"
    DstPort     1
}
Line {
    SrcBlock    "Wind_Direction\n(radius)"
    SrcPort     1
    Points      [160, 0]

```

```

        Branch {
            Points          [0, 530]
            DstBlock        "Azimuth_Wind_Model"
            DstPort          3
        }
        Branch {
            DstBlock        "Altitude_Wind_Model"
            DstPort          4
        }
    }
    Line {
        SrcBlock            "Azimuth_Wind_Model"
        SrcPort              1
        DstBlock            "Mux"
        DstPort              1
    }
    Line {
        SrcBlock            "Azimuth_Wind_Model"
        SrcPort              2
        DstBlock            "Mux"
        DstPort              2
    }
    Line {
        SrcBlock            "Altitude_Wind_Model"
        SrcPort              3
        Points              [75, 0; 0, -70]
        DstBlock            "Mux1"
        DstPort              3
    }
    Line {
        SrcBlock            "Altitude_Wind_Model"
        SrcPort              2
        Points              [75, 0; 0, -15]
        DstBlock            "Mux1"
        DstPort              2
    }
    Annotation {
        Name                "Disturbance"
        Position             [162, 46]
        FontName             "Arial"
        FontSize             28
        FontWeight           "bold"
    }
    Annotation {
        Name                "This Model provide disturbance Input for te"
        "lescope.\nOutputs are in torques (Nm)"
        Position             [86, 79]
        HorizontalAlignment  "left"
    }
}
}
Block {
    BlockType              Mux
    Name                   "Mux"
    Ports                  [4, 1]
    Position                [885, 24, 895, 226]
    ShowName               off
}

```

```

        DisplayOption      "bar"
    }
    Block {
        BlockType      Mux
        Name            "Mux1"
        Ports           [4, 1]
        Position        [890, 285, 900, 470]
        ShowName        off
        DisplayOption    "bar"
    }
    Block {
        BlockType      Mux
        Name            "Mux2"
        Ports           [5, 1]
        Position        [350, 539, 355, 611]
        Orientation     "left"
        ShowName        off
        Inputs          "5"
        DisplayOption    "bar"
    }
    Block {
        BlockType      Scope
        Name            "Scope4"
        Ports           [1]
        Position        [945, 651, 990, 699]
        Location        [8, 56, 696, 860]
        Open            off
        NumInputPorts   "1"
        List {
            ListType      AxesTitles
            axes1          "%<SignalLabel>"
        }
        TimeRange       "12"
        YMin             "-1"
        YMax             "6"
        SaveName         "ScopeData5"
        DataFormat       "StructureWithTime"
        LimitDataPoints  off
    }
    Block {
        BlockType      ToWorkspace
        Name            "To Workplace"
        Position        [975, 104, 1105, 146]
        VariableName     "Azimuth_System_Output"
        MaxDataPoints    "inf"
        SampleTime       "-1"
        SaveFormat       "Array"
    }
    Block {
        BlockType      ToWorkspace
        Name            "To Workplacel"
        Position        [980, 359, 1110, 401]
        VariableName     "Altitude_System_Output"
        MaxDataPoints    "inf"
        SampleTime       "-1"
        SaveFormat       "Array"
    }
}

```

```

Block {
  BlockType      ToWorkspace
  Name           "To Workspace"
  Position       [970, 16, 1105, 54]
  VariableName   "Azimuth_Control_Input"
  MaxDataPoints  "inf"
  SampleTime     "-1"
  SaveFormat     "Array"
}
Block {
  BlockType      ToWorkspace
  Name           "To Workspace1"
  Position       [980, 261, 1115, 299]
  VariableName   "Altitude_Control_Input"
  MaxDataPoints  "inf"
  SampleTime     "-1"
  SaveFormat     "Array"
}
Block {
  BlockType      ToWorkspace
  Name           "To Workspace2"
  Position       [215, 550, 310, 600]
  Orientation     "left"
  VariableName   "Disturbances"
  MaxDataPoints  "inf"
  SampleTime     "-1"
  SaveFormat     "Array"
}
Block {
  BlockType      Constant
  Name           "Wind Direction\n(radius)"
  Position       [85, 740, 115, 770]
  Value          "0"
}
Block {
  BlockType      SubSystem
  Name           "Wind Speed\nGenerator"
  Ports          [0, 1]
  Position       [55, 648, 140, 682]
  TreatAsAtomicUnit  off
  MinAlgLoopOccurrences  off
  RTWSystemCode  "Auto"
  System {
    Name          "Wind Speed\nGenerator"
    Location      [202, 80, 1410, 857]
    Open          off
    ModelBrowserVisibility  on
    ModelBrowserWidth  200
    ScreenColor   "white"
    PaperOrientation  "landscape"
    PaperPositionMode  "auto"
    PaperType      "usletter"
    PaperUnits     "inches"
    ZoomFactor     "100"
  }
  Block {
    BlockType      Sum
    Name           "Add"
  }
}

```

```

Ports          [2, 1]
Position       [365, 77, 395, 108]
InputSameDT    off
OutDataTypeMode "Inherit via internal rule"
OutScaling     "2^-10"
SaturateOnIntegerOverflow off
}
Block {
  BlockType      DiscretePulseGenerator
  Name           "Pulse\nGenerator"
  Ports          [0, 1]
  Position       [140, 163, 185, 197]
  PulseType      "Time based"
  Amplitude      "Wind_amplitude"
  Period         "Wind_period"
  PulseWidth     "80"
  PhaseDelay     "Wind_delay"
}
Block {
  BlockType      RateTransition
  Name           "Rate Transition"
  Position       [250, 64, 290, 106]
}
Block {
  BlockType      Scope
  Name           "Scope1"
  Ports          [1]
  Position       [660, 79, 690, 111]
  Location       [17, 143, 864, 783]
  Open           off
  NumInputPorts  "1"
  List {
    ListType      AxesTitles
    axes1         "%<SignalLabel>"
  }
  TimeRange      "10"
  YMin           "-2"
  SaveName       "ScopeData1"
  DataFormat     "StructureWithTime"
  LimitDataPoints off
}
Block {
  BlockType      UniformRandomNumber
  Name           "Uniform Random\nNumber"
  Position       [55, 69, 85, 101]
  SampleTime     "0.1"
}
Block {
  BlockType      Outport
  Name           "Wind_Speed\n(m/s)"
  Position       [660, 18, 690, 32]
  IconDisplay    "Port number"
  BusOutputAsStruct off
}
Line {
  SrcBlock       "Uniform Random\nNumber"
  SrcPort        1
}

```

```

        DstBlock      "Rate Transition"
        DstPort      1
    }
    Line {
        SrcBlock      "Pulse\nGenerator"
        SrcPort      1
        Points        [135, 0; 0, -80]
        DstBlock      "Add"
        DstPort      2
    }
    Line {
        SrcBlock      "Rate Transition"
        SrcPort      1
        DstBlock      "Add"
        DstPort      1
    }
    Line {
        SrcBlock      "Add"
        SrcPort      1
        Points        [200, 0]
        Branch {
            DstBlock      "Scope1"
            DstPort      1
        }
        Branch {
            Points        [0, -70]
            DstBlock      "Wind_Speed\n(m/s)"
            DstPort      1
        }
    }
}
}
}
Block {
    BlockType      Reference
    Name          "Wind_Switch"
    Ports          [2, 1]
    Position        [305, 657, 335, 693]
    SourceBlock      "simulink/Signal\nRouting/Manual Switch"
    SourceType      "Manual Switch"
    ShowPortLabels    on
    sw              "0"
    action          "0"
}
Line {
    SrcBlock      "Mux"
    SrcPort      1
    DstBlock      "To Workplace"
    DstPort      1
}
Line {
    SrcBlock      "Clock"
    SrcPort      1
    Points        [55, 0]
    Branch {
        Points        [480, 0]
        Branch {
            DstBlock      "Mux"

```

```

    DstPort          1
}
Branch {
    Points          [0, 260]
    DstBlock        "Mux1"
    DstPort          1
}
}
Branch {
Points          [0, 495]
DstBlock        "Mux2"
DstPort          1
}
}
Line {
    SrcBlock        "Azimuth_PID_Controller"
    SrcPort          1
    Points          [130, 0]
    DstBlock        "Azimuth\nInput\nSwitch"
    DstPort          1
}
Line {
    SrcBlock        "Azimuth_Telescope_Structure"
    SrcPort          1
    Points          [20, 0]
    Branch {
Points          [0, 155; -700, 0; 0, -130]
DstBlock        "Azimuth_PID_Controller"
DstPort          2
    }
    Branch {
Points          [30, 0]
Branch {
    Points          [0, 470]
    DstBlock        "Disturbance_Input"
    DstPort          2
}
Branch {
    DstBlock        "Mux"
    DstPort          2
}
}
}
Line {
    SrcBlock        "Azimuth_Telescope_Structure"
    SrcPort          2
    DstBlock        "Mux"
    DstPort          3
}
Line {
    SrcBlock        "Azimuth_Telescope_Structure"
    SrcPort          3
    DstBlock        "Mux"
    DstPort          4
}
Line {
    SrcBlock        "Mux1"

```

```

        SrcPort          1
        DstBlock          "To Workplace1"
        DstPort          1
    }
    Line {
        SrcBlock          "Altitude_Telescope_Structure"
        SrcPort          1
        Points            [5, 0; 10, 0]
        Branch {
            Points        [0, 135; -700, 0; 0, -115]
            DstBlock      "Altitude_PID_Controller"
            DstPort      2
        }
        Branch {
            Points        [15, 0]
            Branch {
                Points    [0, 185]
                DstBlock  "Disturbance_Input"
                DstPort   1
            }
        }
        Branch {
            DstBlock      "Mux1"
            DstPort      2
        }
    }
}
Line {
    SrcBlock          "Altitude_Telescope_Structure"
    SrcPort          2
    DstBlock          "Mux1"
    DstPort          3
}
Line {
    SrcBlock          "Altitude_Telescope_Structure"
    SrcPort          3
    DstBlock          "Mux1"
    DstPort          4
}
Line {
    SrcBlock          "Wind Direction\n(radius)"
    SrcPort          1
    Points            [270, 0]
    Branch {
        Points        [460, 0; 0, -155]
        DstBlock      "Disturbance_Input"
        DstPort      3
    }
    Branch {
        Points        [0, -180]
        DstBlock      "Mux2"
        DstPort      3
    }
}
Line {
    SrcBlock          "Disturbance_Input"
    SrcPort          1
    Points            [-60,          0]

```



```

    Branch {
    Points      [0, -370]
    DstBlock    "Azimuth_Telescope_Structure"
    DstPort     2
    }
    Branch {
    Points      [0, 35]
    DstBlock    "Mux2"
    DstPort     4
    }
}
Line {
    SrcBlock    "Disturbance_Input"
    SrcPort     2
    Points      [-95, 0; 0, -10]
    Branch {
    Points      [0, -170]
    DstBlock    "Altitude_Telescope_Structure"
    DstPort     2
    }
    Branch {
    DstBlock    "Mux2"
    DstPort     5
    }
}
Line {
    SrcBlock    "Mux2"
    SrcPort     1
    DstBlock    "To Workspace2"
    DstPort     1
}
Line {
    SrcBlock    "Wind Speed\nGenerator"
    SrcPort     1
    DstBlock    "Wind_Switch"
    DstPort     1
}
Line {
    SrcBlock    "Wind_Switch"
    SrcPort     1
    Points      [70, 0]
    Branch {
    Points      [420, 0]
    Branch {
    DstBlock    "Scope4"
    DstPort     1
    }
    Branch {
    Points      [0, -45]
    DstBlock    "Disturbance_Input"
    DstPort     4
    }
    }
    Branch {
    Points      [0, -115]
    DstBlock    "Mux2"
    DstPort     2

```

```

    }
}
Line {
    SrcBlock          "Constant"
    SrcPort            1
    Points             [50, 0; 0, -15]
    DstBlock           "Wind_Switch"
    DstPort            2
}
Line {
    SrcBlock          "Azimuth\nInput\nSwitch"
    SrcPort            1
    Points             [30, 0]
    Branch {
DstBlock           "Azimuth_Telescope_Structure"
DstPort            1
    }
    Branch {
Points             [0, -95; 415, 0]
DstBlock           "To Workspace"
DstPort            1
    }
}
Line {
    SrcBlock          "0Input"
    SrcPort            1
    Points             [10, 0; 0, -45]
    DstBlock           "Azimuth\nInput\nSwitch"
    DstPort            2
}
Line {
    SrcBlock          "Altitude\nInput\nSwitch"
    SrcPort            1
    Points             [70, 0]
    Branch {
DstBlock           "Altitude_Telescope_Structure"
DstPort            1
    }
    Branch {
Points             [0, -85]
DstBlock           "To Workspace1"
DstPort            1
    }
}
Line {
    SrcBlock          "Altitude_PID_Controller"
    SrcPort            1
    Points             [100, 0]
    DstBlock           "Altitude\nInput\nSwitch"
    DstPort            1
}
Line {
    SrcBlock          "0Input2"
    SrcPort            1
    Points             [5, 0]
    DstBlock           "Altitude\nInput\nSwitch"
    DstPort            2
}

```

```

    }
    Line {
        SrcBlock      "Azimuth\nInput\nSwitch1"
        SrcPort        1
        DstBlock      "Azimuth_PID_Controller"
        DstPort        1
    }
    Line {
        SrcBlock      "Azimuth_Input"
        SrcPort        1
        DstBlock      "Azimuth\nInput\nSwitch1"
        DstPort        2
    }
    Line {
        SrcBlock      "Altitude\nInput\nSwitch1"
        SrcPort        1
        DstBlock      "Altitude_PID_Controller"
        DstPort        1
    }
    Line {
        SrcBlock      "Altitude_Input"
        SrcPort        1
        DstBlock      "Altitude\nInput\nSwitch1"
        DstPort        2
    }
    Line {
        SrcBlock      "0Input3"
        SrcPort        1
        Points         [5, 0; 0, 30]
        DstBlock      "Azimuth\nInput\nSwitch1"
        DstPort        1
    }
    Line {
        SrcBlock      "0Input1"
        SrcPort        1
        Points         [10, 0; 0, 35]
        DstBlock      "Altitude\nInput\nSwitch1"
        DstPort        1
    }
}
}
}

```

```

%-----
% Azimuth Structure Matrix
%-----
%System Matrix
A = zeros(9,9);
A(1,1)= 0;
A(1,2)= 0;
A(1,3)= 0;
A(1,4)= 0;
A(1,5)= 1;
A(1,6)= 0;
A(1,7)= 0;
A(1,8)= 0;
A(1,9)= 0;

A(2,1)= 0;
A(2,2)= 0;
A(2,3)= 0;
A(2,4)= 0;
A(2,5)= 0;
A(2,6)= 1;
A(2,7)= 0;
A(2,8)= 0;
A(2,9)= 0;

A(3,1)= 0;
A(3,2)= 0;
A(3,3)= 0;
A(3,4)= 0;
A(3,5)= 0;
A(3,6)= 0;
A(3,7)= 1;
A(3,8)= 0;
A(3,9)= 0;

A(4,1)= 0;
A(4,2)= 0;
A(4,3)= 0;
A(4,4)= 0;
A(4,5)= 0;
A(4,6)= 0;
A(4,7)= 0;
A(4,8)= 1;
A(4,9)= 0;

A(5,1)= -(K_f)/J_mo;
A(5,2)= (K_f)/J_mo;
A(5,3)= 0;
A(5,4)= 0;
A(5,5)= -(B_mo+B_f)/J_mo;
A(5,6)= B_f/J_mo;
A(5,7)= 0;

```

```

A(5,8)= 0;
A(5,9)= 1/(Ka*J_mo);

A(6,1)= K_f/J_am;
A(6,2)= -(K_f+K_o+K_m)/J_am;
A(6,3)= K_o/J_am;
A(6,4)= K_m/J_am;
A(6,5)= B_f/J_am;
A(6,6)= -(B_f+B_o+B_m)/J_am;
A(6,7)= B_o/J_am;
A(6,8)= B_m/J_am;
A(6,9)= 0;

A(7,1)= 0;
A(7,2)= K_o/J_o;
A(7,3)= -K_o/J_o;
A(7,4)= 0;
A(7,5)= 0;
A(7,6)= B_m/J_o;
A(7,7)= -B_o/J_o;
A(7,8)= 0;
A(7,9)= 0;

A(8,1)= 0;
A(8,2)= K_m/J_m;
A(8,3)= 0;
A(8,4)= -K_m/J_m;
A(8,5)= 0;
A(8,6)= B_m/J_m;
A(8,7)= 0;
A(8,8)= -B_m/J_m;
A(8,9)= 0;

A(9,1)= 0;
A(9,2)= 0;
A(9,3)= 0;
A(9,4)= -1/(Ka*L_e);
A(9,5)= 0;
A(9,6)= 0;
A(9,7)= 0;
A(9,8)= 0;
A(9,9)= -R_e/L_e;

%Input Matrix
B = zeros(9,1);
B(1,1)= 0;
B(2,1)= 0;
B(3,1)= 0;
B(4,1)= 0;
B(5,1)= 0;
B(6,1)= 0;
B(7,1)= 0;
B(8,1)= 0;
B(9,1)= 1/L_e;

%Disturbance Matrix

```

```

E = zeros(9,2);
E(1,1)= 0;
E(1,2)= 0;
E(2,1)= 0;
E(2,2)= 0;
E(3,1)= 0;
E(3,2)= 0;
E(4,1)= 0;
E(4,2)= 0;
E(5,1)= 0;
E(5,2)= 0;
E(6,1)= 0;
E(6,2)= 0;
E(7,1)= 1/J_o;
E(7,2)= 0;
E(8,1)= 0;
E(8,2)= 1/J_m;
E(9,1)= 0;
E(9,2)= 0;

```

```

%Output Matrix
C = zeros(3,9);
C(1,1)= 1;
C(1,2)= 0;
C(1,3)= 0;
C(1,4)= 0;
C(1,5)= 0;
C(1,6)= 0;
C(1,7)= 0;
C(1,8)= 0;
C(1,9)= 0;

```

```

C(2,1)= 0;
C(2,2)= 0;
C(2,3)= 0;
C(2,4)= 0;
C(2,5)= 1;
C(2,6)= 0;
C(2,7)= 0;
C(2,8)= 0;
C(2,9)= 0;

```

```

C(3,1)= -1*L_OTA;
%C(3,1)= 0;
C(3,2)= 0*L_OTA;
C(3,3)= -L_OTA;
C(3,4)= 2*L_OTA;
C(3,5)= 0;
C(3,6)= 0;
C(3,7)= 0;
C(3,8)= 0;
C(1,9)= 0;

```

```

%Direct Transmittance Matrix
D = zeros(3,1);

```

```

%-----
% Altitude Structure Matrix
%-----
%System Matrix
A_a = zeros(9,9);
A_a(1,1)= 0;
A_a(1,2)= 0;
A_a(1,3)= 0;
A_a(1,4)= 0;
A_a(1,5)= 1;
A_a(1,6)= 0;
A_a(1,7)= 0;
A_a(1,8)= 0;
A_a(1,9)= 0;

A_a(2,1)= 0;
A_a(2,2)= 0;
A_a(2,3)= 0;
A_a(2,4)= 0;
A_a(2,5)= 0;
A_a(2,6)= 1;
A_a(2,7)= 0;
A_a(2,8)= 0;
A_a(2,9)= 0;

A_a(3,1)= 0;
A_a(3,2)= 0;
A_a(3,3)= 0;
A_a(3,4)= 0;
A_a(3,5)= 0;
A_a(3,6)= 0;
A_a(3,7)= 1;
A_a(3,8)= 0;
A_a(3,9)= 0;

A_a(4,1)= 0;
A_a(4,2)= 0;
A_a(4,3)= 0;
A_a(4,4)= 0;
A_a(4,5)= 0;
A_a(4,6)= 0;
A_a(4,7)= 0;
A_a(4,8)= 1;
A_a(4,9)= 0;

A_a(5,1)= -(K_o_a+K_m_a)/J_mo_a;
A_a(5,2)= 0;
A_a(5,3)= K_o_a/J_mo_a;
A_a(5,4)= K_m_a/J_mo_a;
A_a(5,5)= -(B_mo_a+B_o_a+B_m_a)/J_mo_a;
A_a(5,6)= B_mo_a/J_mo_a;
A_a(5,7)= B_o_a/J_mo_a;
A_a(5,8)= B_m_a/J_mo_a;
A_a(5,9)= 1/(Ka_a*J_mo_a);

```

```

A_a(6,1)= 0;
A_a(6,2)= -K_f_a/J_f_a;
A_a(6,3)= 0;
A_a(6,4)= 0;
A_a(6,5)= B_mo_a/J_f_a;
A_a(6,6)= -(B_mo_a+B_f_a)/J_f_a;
A_a(6,7)= 0;
A_a(6,8)= 0;
A_a(6,9)= -1/(Ka_a*J_f_a);

```

```

A_a(7,1)= K_o_a/J_o_a;
A_a(7,2)= 0;
A_a(7,3)= -K_o_a/J_o_a;
A_a(7,4)= 0;
A_a(7,5)= B_o_a/J_o_a;
A_a(7,6)= 0;
A_a(7,7)= -B_o_a/J_o_a;
A_a(7,8)= 0;
A_a(7,9)= 0;

```

```

A_a(8,1)= K_m_a/J_m_a;
A_a(8,2)= 0;
A_a(8,3)= 0;
A_a(8,4)= -K_m_a/J_m_a;
A_a(8,5)= B_m_a/J_m_a;
A_a(8,6)= 0;
A_a(8,7)= 0;
A_a(8,8)= -B_m_a/J_m_a;
A_a(8,9)= 0;

```

```

A_a(9,1)= 0;
A_a(9,2)= 0;
A_a(9,3)= 0;
A_a(9,4)= 0;
A_a(9,5)= -1/(Ka_a*L_e_a);
A_a(9,6)= 1/(Ka_a*L_e_a);
A_a(9,7)= 0;
A_a(9,8)= 0;
A_a(9,9)= -R_e_a/L_e_a;

```

```

%Input Matrix

```

```

B_a = zeros(9,1);
B_a(1,1)= 0;
B_a(2,1)= 0;
B_a(3,1)= 0;
B_a(4,1)= 0;
B_a(5,1)= 0;
B_a(6,1)= 0;
B_a(7,1)= 0;
B_a(8,1)= 0;
B_a(9,1)= 1/L_e_a;

```

```

%Disturbance Matrix

```

```

E_a = zeros(9,5);
E_a(1,1)= 0;
E_a(1,2)= 0;

```



```

E_a(1,3)= 0;
E_a(1,4)= 0;
E_a(1,5)= 0;

E_a(2,1)= 0;
E_a(2,2)= 0;
E_a(2,3)= 0;
E_a(2,4)= 0;
E_a(2,5)= 0;

E_a(3,1)= 0;
E_a(3,2)= 0;
E_a(3,3)= 0;
E_a(3,4)= 0;
E_a(3,5)= 0;

E_a(4,1)= 0;
E_a(4,2)= 0;
E_a(4,3)= 0;
E_a(4,4)= 0;
E_a(4,5)= 0;

E_a(5,1)= 0;
E_a(5,2)= 0;
E_a(5,3)= 0;
E_a(5,4)= 0;
E_a(5,5)= 0;

E_a(6,1)= 0;
E_a(6,2)= 0;
E_a(6,3)= 1/J_f_a;
E_a(6,4)= 0;
E_a(6,5)= 0;

E_a(7,1)= 1/J_o_a;
E_a(7,2)= 0;
E_a(7,3)= 0;
E_a(7,4)= 1/J_o_a;
E_a(7,5)= 0;

E_a(8,1)= 0;
E_a(8,2)= 1/J_m_a;
E_a(8,3)= 0;
E_a(8,4)= 0;
E_a(8,5)= 1/J_m_a;

E_a(9,1)= 0;
E_a(9,2)= 0;
E_a(9,3)= 0;
E_a(9,4)= 0;
E_a(9,5)= 0;

%Output Matrix
C_a = zeros(3,9);
C_a(1,1)= 1;

```

```

C_a(1,2)= 0;
C_a(1,3)= 0;
C_a(1,4)= 0;
C_a(1,5)= 0;
C_a(1,6)= 0;
C_a(1,7)= 0;
C_a(1,8)= 0;
C_a(1,9)= 0;

C_a(2,1)= 0;
C_a(2,2)= 0;
C_a(2,3)= 0;
C_a(2,4)= 0;
C_a(2,5)= 1;
C_a(2,6)= 0;
C_a(2,7)= 0;
C_a(2,8)= 0;
C_a(2,9)= 0;

C_a(3,1)= -1*L_OTA;
%C_a(3,1)= 0;
C_a(3,2)= 1*L_OTA;
C_a(3,3)= -L_OTA;
C_a(3,4)= 2*L_OTA;
C_a(3,5)= 0;
C_a(3,6)= 0;
C_a(3,7)= 0;
C_a(3,8)= 0;
C_a(1,9)= 0;

%Direct Transmittance Matrix
D_a = zeros(3,1);

```

```

%-----
% Pre-run parameters
%
%Chi-Yeh Hsu
%
% run it before use telescope model
%-----

%-----
% Plot Parameters
%-----
%-----
% Matlab Parameters
%-----
t_end = 5;
%-----
% Control Input Parameters
%-----
Azimuth_initial = 0;
Azimuth_final   = pi/12;
Azimuth_stepTime= 0.5;

Altitude_initial = 0;
Altitude_final   = pi/12;
Altitude_stepTime= Azimuth_stepTime;

%-----
% Wind Parameters
%-----
Wind_amplitude = 5;
Wind_period = 10;
Wind_delay = 6;
%-----
% Environmental Parameters
%-----
p_air = 1.25;          % density of air (kg/m^3)
g = 0;                 % Gravitational acceleration(m/s^2)
%-----
% Initial States
%-----
%Azimuth Initial State (11 States)
Theta_J = 0;
Theta_Jam= Theta_J;
Theta_JO = Theta_J;
Theta_Jm = Theta_J;
Omega_J = 0;
Omega_Jam= Omega_J;
Omega_JO = Omega_J;
Omega_Jm = Omega_J;
i_L = 0;

X_ini = [Theta_J Theta_Jam Theta_JO Theta_Jm Omega_J Omega_Jam Omega_JO
Omega_Jm i_L];

```

```

%Altitude Initial State (9 States)
Theta_J_a = 0;
Theta_Jf_a = 0;
Theta_JO_a = Theta_J_a;
Theta_Jm_a = Theta_J_a;
Omega_J_a = 0;
Omega_Jf_a = Omega_J_a;
Omega_JO_a = Omega_J_a;
Omega_Jm_a = Omega_J_a;
i_L_a = 0;

X_a_ini = [Theta_J_a Theta_Jf_a Theta_JO_a Theta_Jm_a Omega_J_a
Omega_Jf_a Omega_JO_a Omega_Jm_a i_L_a];
%-----
% Control Parameters
%-----
Kp = 24;
Kd = 8;
Ki = 0;

Kp_a = 5;
Kd_a = 1.6;
Ki_a = 0;

```

Appendix E: Matlab Plot Functions

```
%-----  
% Plot function  
% this function plots azimuth motion result  
%-----  
clc  
%-----  
% Plot Parameters  
%-----  
Y_min = min(Azimuth_System_Output(:,2))-  
max(Azimuth_System_Output(:,2))/10;  
Y_max =  
max(Azimuth_System_Output(:,2))+max(Azimuth_System_Output(:,2))/10;  
X_l_min = min(Azimuth_System_Output(:,4))-  
max(Azimuth_System_Output(:,4))/10;  
X_l_max =  
max(Azimuth_System_Output(:,4))+max(Azimuth_System_Output(:,4))/10;  
  
P_f = Azimuth_System_State(:,2)-Azimuth_System_State(:,1);  
P_o = Azimuth_System_State(:,3)-Azimuth_System_State(:,2);  
P_m = Azimuth_System_State(:,4)-Azimuth_System_State(:,2);  
% P_f = Azimuth_System_State(:,2);  
% P_o = Azimuth_System_State(:,3);  
% P_m = Azimuth_System_State(:,4);  
C_input = Azimuth_Control_Input(:,1);  
  
% P_f_min = min(P_f)-max(P_f)/10;  
% P_f_max = max(P_f)+max(P_f)/10;  
% P_o_min = min(P_o)-max(P_o)/10;  
% P_o_max = max(P_o)+max(P_o)/10;  
% P_m_min = min(P_m)-max(P_m)/10;  
% P_m_max = max(P_m)+max(P_m)/10;  
% C_input_min = min(C_input)-max(C_input)/10;  
% C_input_max = max(C_input)+max(C_input)/10;  
  
% Vib_max = 1*10^-5;  
% Vib_min = -Vib_max;  
%-----  
% Plot Function  
%-----  
figure(1);  
subplot(3,2,1)  
plot (Azimuth_System_Output(:,1), Azimuth_System_Output(:,2), '-');  
% Default line type is solid  
grid on;  
title (['Telescope Azimuth Rotation Position, Kp=',num2str(Kp),',  
Kd=',num2str(Kd),', Ki=',num2str(Ki)]);  
% You must have graph title  
xlabel ('Time [sec]');  
ylabel ('Position [Radius]');  
axis tight
```

```

subplot(3,2,3)
plot (Azimuth_System_Output(:,1), Azimuth_System_Output(:,4), '-k');
% Default line type is solid
grid on;
title ('Azimuth Image Sensor Signal Displacement');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Position [M]');
axis tight

subplot(3,2,5)
plot (Azimuth_System_Output(:,1), C_input, 'r');
% Default line type is solid
grid on;
title ('Azimuth Control Input');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Voltage (V)');
axis tight

subplot(3,2,2)
plot (Azimuth_System_Output(:,1), P_f, '-');
% Default line type is solid
grid on;
title ('Azimuth Fork Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight

subplot(3,2,4)
plot (Azimuth_System_Output(:,1), P_o, '-');
% Default line type is solid
grid on;
title ('Azimuth OTA Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight

subplot(3,2,6)
plot (Azimuth_System_Output(:,1), P_m, '-');
% Default line type is solid
grid on;
title ('Azimuth Mirror Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight

```

```

%-----
% Plot function
% This function plots altitude motion result
%-----
clc
%-----
% Plot Parameters
%-----
Y_min = min(Altitude_System_Output(:,2))-
max(Altitude_System_Output(:,2))/10;
Y_max =
max(Altitude_System_Output(:,2))+max(Altitude_System_Output(:,2))/10;
X_l_min = min(Altitude_System_Output(:,4))-
max(Altitude_System_Output(:,4))/10;
X_l_max =
max(Altitude_System_Output(:,4))+max(Altitude_System_Output(:,4))/10;

P_f = Altitude_System_State(:,2);
P_o = Altitude_System_State(:,3)-Altitude_System_State(:,1);
P_m = Altitude_System_State(:,4)-Altitude_System_State(:,1);
C_input = Altitude_Control_Input(:,1);

% P_f_min = min(P_f)-max(P_f)/10;
% P_f_max = max(P_f)+max(P_f)/10;
% P_o_min = min(P_o)-max(P_o)/10;
% P_o_max = max(P_o)+max(P_o)/10;
% P_m_min = min(P_m)-max(P_m)/10;
% P_m_max = max(P_m)+max(P_m)/10;
% C_input_min = min(C_input)-max(C_input)/10;
% C_input_max = max(C_input)+max(C_input)/10;

Vib_max = 1*10^-3;
Vib_min = -Vib_max;
%-----
% Plot Function
%-----
figure(2);
subplot(3,2,1)
plot (Altitude_System_Output(:,1), Altitude_System_Output(:,2),'-');
% Default line type is solid
grid on;
title (['Telescope Altitude Rotation Position, Kp=',num2str(Kp_a),',
Kd=',num2str(Kd_a),', Ki=',num2str(Ki_a)]);
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Position [Radius]');
axis tight

subplot(3,2,3)
plot (Altitude_System_Output(:,1), Altitude_System_Output(:,4),'-k');
% Default line type is solid
grid on;
title ('Altitude Image Sensor Signal Displacement');
% you must have graph title
xlabel ('Time [sec]');

```

```

ylabel ('Position [M]');
axis tight

subplot(3,2,5)
plot (Altitude_System_Output(:,1), C_input, 'r');
% Default line type is solid
grid on;
title ('Control Input');
% you must have graph title
xlabel ('Time [sec]');
ylabel ('Voltage (V)');
axis tight

subplot(3,2,2)
plot (Altitude_System_Output(:,1), P_f, '-');
% Default line type is solid
grid on;
title ('Fork Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight

subplot(3,2,4)
plot (Altitude_System_Output(:,1), P_o, '-');
% Default line type is solid
grid on;
title ('OTA Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight

subplot(3,2,6)
plot (Altitude_System_Output(:,1), P_m, '-');
% Default line type is solid
grid on;
title ('Mirror Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight

```


Appendix F: OTA Calculation

```
%-----  
% this function calculates Beam properties of the OTA  
%  
% Chi-Yeh Hsu  
%-----  
clc  
clear  
%Steel Beam  
l = 450/1000;           %m  
w = 38.1/1000;          %m  
h = 3.175/1000;         %m  
E = 200 *10^9;          %N/m^2  
  
density = 7.85*10^3;     %kg/m^3  
m = l*w*h*density        %kg  
  
%telescope length  
L_OTA = 0.45;           %m  
m_OTA = 0.6;            %kg OTA  
m_mm = 0.25;            %kg mirror mount  
m_sen = 0.15;           %kg sensor  
m_m = 0.5;              %kg mirror  
  
I_total =  
m_sen*(L_OTA*0.7)^2+m_m*(L_OTA*0.3)^2+m_OTA*(L_OTA*0.7*0.5)^2+m_mm*(L_OTA*0.3*0.5)^2 %kg*m^2  
I_OTA = m_sen*(L_OTA*0.7)^2+m_OTA*(L_OTA*0.7*0.5)^2  
I_mm = m_m*(L_OTA*0.3)^2+m_mm*(L_OTA*0.3*0.5)^2  
  
%spring constant  
  
I_o = w*(h^3)/12  
K =3*E*I_o/(l)  
K_1 = 2*3*E*I_o/(l*0.7)  
K_2 = 2*3*E*I_o/(l*0.3)
```

Appendix G: System Parameters

```
clear
clc
%-----
% Structure Parameters
%-----
%Electric System-----
%
%Azimuth System
R_e  = 0.37;           %Internal Resistance of Motor
L_e  = 0.56/1000;      %Internal Inductance of motor
Ka = 30.58;           %Motor Constant (A/Nm or rad/V)
%Altitude System
R_e_a  = 0.37;
L_e_a  = 0.56/1000;
Ka_a = 30.58;

%Telescope Fork-----
%
%Azimuth System
B_f = 0.8;           %Friction (damping) of fork
K_f = 2000;          %Spring constant of fork
J_f = 0.149;         %Inertia of fork
L_Fork = 0.2;        %Length of Fork (m)
%Altitude System
B_f_a = 0.8;
K_f_a = 2000;
J_f_a = 0.149;
A_Fork_a = 0.01;

%Motor System-----
%
%Azimuth System
B_mo = 1.2*10^-5;     %Friction (damping) between Transmission and base
                        % (bearing)
J_mo = 2.1*10^-5;     %Inertia of Section 1(motor and transmission)A
J_am = J_mo+J_f;
%Altitude System
B_mo_a = 1.2*10^-5;
J_mo_a = 2.1*10^-5;

%Mirror Mount-----
%
%Azimuth System
B_m = 0.2;           %Friction (damping) of mirror mount
K_m = 903;           %Spring constant of node mirror mount
J_m = 0.0103;        %Inertia of mirror mount
L_mm  = 0.05;        %Length of Mirror Mount (m)
A_mm  = 0.0075;      %Area of Mirror Mount (m^2)
%Altitude System
B_m_a = 0.2;
K_m_a = 903;
J_m_a = 0.0103;
```

```

A_mm_a = 0.0075;    % Area of mirror mount (m^2)
m_mm = 0.75;        %Mirror Mount Mass (kg)

%OTA-----
%
%Azimuth System
B_o = 0.1;          %Friction (damping) of node 1(OTA)
K_o = 387;          %Spring constant of node 1(OTA)
J_o = 0.0298;       %Inertia of OTA
L_OTA = 0.45;       %Length of OTA
A_OTA = 0.06;       % Area of OTA (m^2)
%Altitude System
B_o_a = 0.1;
K_o_a = 387;
J_o_a = 0.0298;
A_OTA_a = 0.06;     % Area of OTA (m^2)
m_OTA= m_mm*L_mm/L_OTA;    %OTA mass (kg)

```